Commodore INFO

C-16 machinetaal

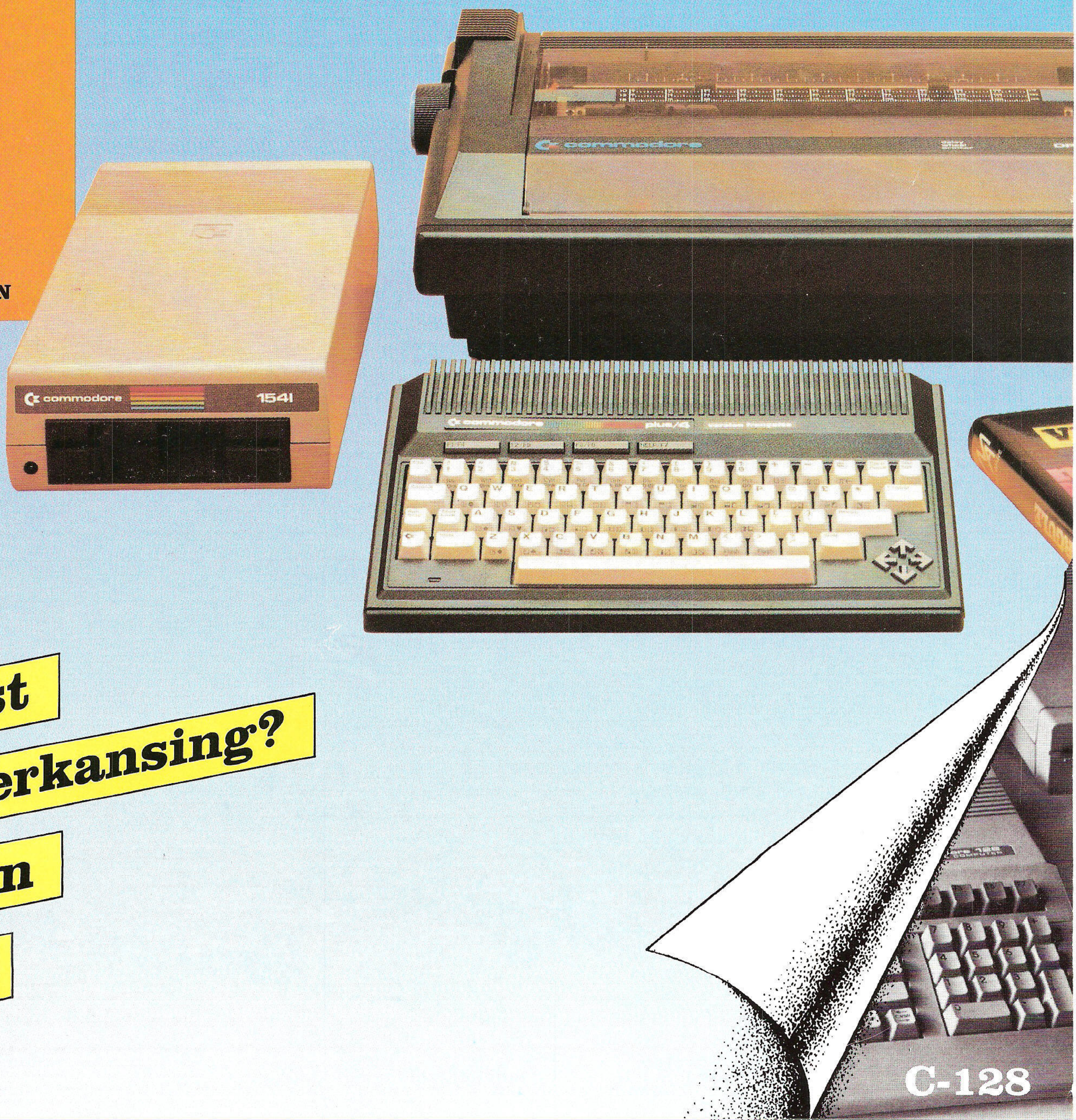
Invoerroutines

64 machinetaal

ONAFHANKELIJK BLAD VOOR COMMODORE GEBRUIKERS   PRIJS ƒ 6,50/Bfr. 130   JAARGANG 2, No. 8, 1985

LISTINGS

CHECKSUM
HIT THE BALL
SJOELEN 64
COMMODORE-SHOW
CATCH THE BALL
FACULTEIT
BEELDEFFECT
TAFELS
KETTING 64
MACHTSVERHEFFEN
BLAUWE BONEN
GELDWEG
VIC LISTSTOP
VIC ONDERSTEBOVEN



C-128 test

Plus-4 herkansing?

Activision

Utilities

C-128

Redactioneel

Commodore-Info noemt zich een onafhankelijk computerblad. Daarmee bedoelen we, dat we geen banden hebben met bv. Commodore en dat de lezer er op kan vertrouwen, dat onze informatie over Commodore dus kritisch en onafhankelijk van de commerciële belangen van Commodore is. Zoals we al eerder hebben gezegd, dat bedrijf is net als alle andere een gewaardeerd adverteerder, maar veel verder gaat het niet. Commodore verschijnt bijvoorbeeld niet op onze Commodore-Info beurzen, en stelt zich tevreden met het regelen van wat 128's, die wij dan aan het publiek kunnen tonen.

Die onafhankelijkheid wordt in het volgend nummer nog duidelijker, want we kregen net voor het ter perse gaan van dit blad te horen, dat Commodore alle advertentiecontracten heeft opgezegd.
Dat vinden we jammer, niet alleen om commerciële redenen, maar vooral omdat we vrezen, dat het bedrijf zich als vanouds wel vriendelijk opstelt tegenover nieuwe klanten, maar de bestaande niet meer ziet zitten. En lezers van Commodore-Info zijn tenslotte meest bestaande klanten, bezitters van machines die service, informatie en support nodig hebben. Computerbladen als het onze hebben daarbij ook een funktie, maar het is in eerste instantie de verantwoordelijkheid van Commodore zelf.

Jammer, we hopen dat Commodore in de toekomst weer als adverteerder terugkomt en dan met name het brede aanbod aan randapparatuur en software onder de aandacht wil brengen.

Luc Sala

Inhoud van dit nummer

Datakolom 5
Luc Sala over computerfrustraties

Seizoensuitverkoop 7
Nieuws van de computerbeurzen

Nieuws 9

Programmeerbare Karakters 11
Het maken van een eigen tekenset en standaardfiguurtjes met voorbeeldprogramma.

Gebruikersgroepen 15
Nuttige adressen

Software 16
Nieuws over programma's en bedrijven.

Activision 21
Een softwarehuis vol knallers, hun nieuwe produkten alvast bekeken.

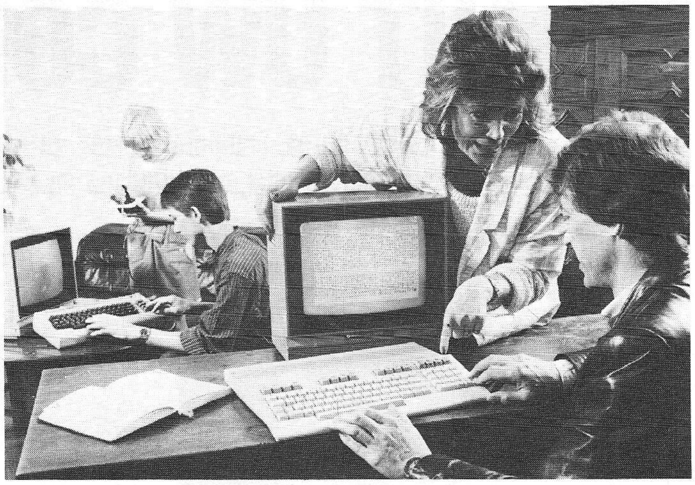
Tekenstaaf 28
Hardware: Een alternatief voor de joystick met betere stuureigenschappen.

Basic Miniatuurtjes 30
Nico Baaijens heeft wat meer keus uit betere inzendingen, dus weer heel wat goede kortere programma's.

Listing PRINT-OUT

Weer erg veel pagina's listings met:

Checksum	38
Hit the Ball	39
Sjoelen 64	39
Commodore-Show	40
Catch the Ball	51
Faculteit	52
Beeldeffect	52
Tafels	53
Ketting 64	54
Machtsverheffen	55
Blauwe bonen	56
Geldweg	56
Vic Liststop	59
Vic Ondersteboven	59



Commodore bureaumachine

Utilities 33
Vier bladzijden met hulpprogramma's, voor wie zijn Commodore optimaal wil benutten.

De 128 onder vuur 44
We probeerden de nieuwste Commodore CBM 128 en Jan Bodzinga rapporteert op zijn kritische manier, maar wel gebaseerd op "hands-on" ervaringen.

Vragenrubriek 60
Jan Bodzinga beantwoordt weer uw probleemvragen.

Lettervergroting 65
Het maken van een blow-up van letters of schermbeelden; met demoprogramma

Missers 68
Fouten in en commentaar op vorige listings.

Soft Wir-War 70
Bert Tier's exclusieve strip

Boekenrubriek 72

Machinetaal cursus deel 6 75
Sjoerd Bakker met een aflevering over de SID en wat je daar aan muziek mee kunt maken.

Invoer routine 79
Maak het invoeren van variabelen tot een eenvoudig karweitje met dit artikel.

C-16 Machinetaal 85
Een diepgravende serie over de mogelijkheden van de C-16

COMMODORE-INFO
Jaargang 2, no. 8
Dit blad wordt uitgegeven door:
SAC, PB 112, 1260 AC te Blaricum,
NL. tel. 02152-65695.
Commodore-Info is een onafhankelijk blad en verschijnt tenminste 10 x per jaar.
Abonnement ƒ 55,– per jaar op giro 1585491 tnv SAC Blaricum.
Abonnementenadministratie:
Den Texstraat 5A,
1017 XW Amsterdam
Postbus 5570, 1007 AN te Amsterdam
Tel. 020-273198

Voor Belgie: B.B.Lambert,
(ƒ 55,– of Bfr 1100)
banknr. 310050602562
tnv SAC Blaricum.

Distributie: Betapress Gilze
voor België: AMP Brussel

Redactie:

Ir. L. Sala	hoofdredacteur
Drs. J. Taverne	uitgever
Ing. V. Sala	logistiek
J. Bodzinga	techniek
B. Munniksma	onderwijs
R. Ramdjanamsingh	hardware
K. van der Vlies	reportage
Rob vd Heuvel	listings
Bert Tier	cartoons/strip
Ben van Mierlo	illustráties

Druk: Drukkerij Verweij B.V. - Mijdrecht

ISSN 0169-3085



LUC SALA'S DATAKOLOM

Velen zijn hun computerhobby begonnen met de aanschaf van een micro, die naar zij toen dachten, binnen de begroting viel. Blij met hun aanwinst wilden ze daarmee dan iets gaan doen, en kwamen er dan al rap achter dat ze dan nog wel iets extra's moesten aanschaffen. Een cassetterecorder, monitor, aansluitkabels, niet te vergeten software, toolkits, kopafstelsets, drives, modems, meer kabels, meer software, na een paar jaar staat er voor vele duizenden guldens.

Het verhaal klinkt u bekend in de oren, want vrijwel iedereen is in dit gat gevallen en is dan misschien wel trots op zijn nu complete systeem, maar toch ook een beetje spijtig. Want die leuke vakantie, die nieuwe auto of het kerstkado voor de kinderen, dat staat daar nu wel in de hoek.

Natuurlijk hebben de fabrikanten er belang bij, om spullen te leveren, waarbij de herhalings- of uitbreidingsaankopen er al zitten ingebakken. Dat verhoogt de omzet en vooral de winkeliers zijn er erg blij mee, want de klant komt terug en koopt nog iets. Er worden leuke zaken gedaan in cassetterecorders, kabeltjes en dergelijke zaken.

De verhouding tussen de computer zelf en wat er bij komt is bij systemen als de C-16 wel helemaal zoek, naar je gevoel staat 159 gulden voor de cassetterecorder in geen verhouding meer met de prijs van f 299,- voor het ding zelf.

Nu zijn er soms opflakkeringen van gezond verstand bij de hardware-fabrikanten en dan komen ze voor een tijdje met een totaalaanbieding, een systeem plus software plus documentatie plus cassetterecorder. Dat lijkt dan prima, maar bedenk wel, dat er vaak wel een addertje onder het gras zit. De aansluitingen zijn meestal zo weinig standaard, dat men alleen met een totaalpakket de klanten en detaillisten nog een beetje tevreden kan houden.

Amstrad is een voorbeeld van zo'n totaal-aanbod. Bij elkaar een leuke set, maar je moet wel in één keer alles kopen en nog eens een andere monitor of zoiets kiezen is er niet bij.

Want dat wordt je natuurlijk voor de voeten geworpen, als je klaagt over de opdeling van het systeem in allerlei componenten. Je hebt zo toch meer keus, je kunt je eigen systeem samenstellen, nietwaar. Klinkt aardig, maar in de praktijk heeft (bij de 64) minder dan 5% een afwijkende diskdrive en ook de datasette is vrijwel universeel, de alternatieven zijn wel goedkoper, maar ook niet zo erg betrouwbaar.

Een gevaar van deze fragmentatie is, dat men stapje voor stapje meegezogen wordt naar een systeem, waar men uiteindelijk toch niet tevreden mee is. Want de basismachine, waarmee gestart werd, is te zwak om de hele batterij randapparatuur adequaat te behandelen, de geheugenruimte is te krap, de drivecapaciteit onvoldoende, de laadsnelheid irritant, men groeit uit de machine.

Gelukkig maar, anders zaten we allemaal nog met 1K op onze ZX-80 te prutsen, maar wel jammer van het geld. En er valt weinig tegen te doen, er worden nu eenmaal geen goedkope basismachines verkocht, die voldoende standaard zijn en toch krachtig genoeg om echt doorgroeien mogelijk te maken. Ik kan slechts waarschuwen voor de beperkingen. Wie denkt, dat hij later wel iets meer wil gaan doen met zijn micro, kan beter direkt een stapje omhoog doen en een 128 of PC-10 aanschaffen.

Er is natuurlijk ook de mogelijkheid, dat men eerst eens wil experimenteren. Dan is een goedkoop micro'tje natuurlijk leuk als eerste oefening, maar wel met de uiterste beperking wat betreft extra's. Zo gauw de neiging opkomt, om te gaan uitbreiden, is het waarschijnlijk beter, het ding in de kast te zetten of weg te geven en echt opnieuw te beginnen. Dat lijkt op het weggooien van geld, maar in wezen is de aanschafprijs van de basismachine vergeleken met die van een compleet systeem toch gering. Een paar honderd of zelfs duizend gulden voor hardware is wel veel geld, maar wanneer men bv. voor professionele pakketten als Wordstar 2000, Lotus 1-2-3 of een boekhoudpakket toch met prijzen van duizend gulden of meer moet rekenen, valt dat dus wel mee.

Het alternatief is een topzware installatie, die misschien net alles doet, wat men wil, maar dan wel met meer kabels, meer storingsbronnen en toch maar beperkte mogelijkheden. ●

Op diverse Europese computerbeurzen zijn we aan het speuren geweest naar nieuws van en over Commodore. Onze redacteuren waren in Berlijn, Londen en Paris, waar het najaarsseizoen der computerfreaks van start ging. OOk op de Commodore-Info computerdag in Den Haag, waar 7500 bezoekers hun slag kwamen slaan, werd duidelijk waar we naar toe gaan. De hooggespannen verwachtingen zijn voorbij, het is meer een kwestie van overleven geworden.

SEIZOENS-UITVERKOOP



Computerbeurzen zijn niet meer wat ze een paar jaar geleden waren, toen het enthousiasme van de gebruikers en de reclamebudgetten van de leveranciers geen grenzen kenden. Toen kondigde men met groot gemak allerlei "Vaporware" aan, produkten die (nog) niet bestonden, smeet er flink wat promotie tegenaan en noteerde bestellingen. Dat is nu wel even wat anders, het publiek kijkt, vraagt, kijkt nog eens en gaat dan aan anderen vragen of het allemaal wel te vertrouwen is. De CBM 128, nu toch al meer dan 9 maanden in het nieuws, werd overal wel getoond, maar wanneer men een exemplaar kon kopen en meenemen bleef steeds een vraag. Misschien begin october, zegt Commodore, maar misschien ook later.

Wie nog een diskdrive, printer of computer had willen hebben, was op 14 september in de Houtrusthallen aardig goedkoop uitgeweest. Daar werd door bijna vijftig standhouders op grote schaal opgeruimd. De dozen met apparatuur gingen bij honderden de deur uit, we waren er zelf verbaasd over. De prijzen leken overigens lager dan ze waren, Commodore heeft zelf namelijk door het samenstellen van een combinatiepakket Easyscript/1541/MPS 801 voor de Engelse markt de prijsdumping veroorzaakt. Die pakketten doken namelijk in grote aantallen in ons land op, maar de handelaren bedachten al snel, dat het gesplitste pakket meer op zou brengen dan de combinatie. Dus liet een slimme vogel gauw nieuwe dozen drukken, daar werden de drives uit de opgedeelde pakketten ingestopt en los verkocht. Veel lager dan de officiele Commodoreprijs, maar nog steeds met een redelijke marge voor de winkelier.

C-16 revival

Tegen de weggeefprijs, waarvoor je tegenwoordig een C-16 startpakket kunt kopen, is dat zonder enige twijfel het computersnoepje van de week. Minder dan driehonderd gulden, een beetje computercursus kost tegenwoordig meer. Er werden er ook in Den Haag weer tientallen verkocht en uit de reakties en ook uit de vraag naar bv. het Infolist MCT 64 Ramboard voor de C-16 kunnen we toch afleiden, dat de C-16 en de Plus4 via de achterdeur toch populaire machines aan het worden zijn.
Dat vinden ze schijnbaar ook in andere landen, want met name de Plus-4 duikt toch weer op en verdient qua prijs/prestatieverhouding ook wel degelijk een plaatsje onder de zon. Elders in dit nummer gaan we nog eens uitgebreid in op de mogelijkheden van deze machine.

Funkausstellung

Op de Berlijns Funkaussttellung, die een beetje op de Firato lijkt, was Commodore opvallend afwezig en liet de vloer daarmee eigenlijk aan de MSX machines. De software-huizen, en daaronder was ook Aackosoft, trokken zich daar niet veel van aan en brachten wel veel software voor de C-64. En de honderduizenden Berlijnse jongeren bevolkten dan ook die stands, probeerden de software uit en bepaalden grotendeels de sfeer van de tentoonstelling. Ons viel op, dat er van duitse bodem zo weinig leuke spelletjes of creatieve toepassingen komen, men houdt zich daar liever bij serieuze programmeertalen, zoals het grote aanbod van Data-Becker. Op hun C-Compiler komen we nog terug. Allemaal degelijk, goed gedocumenteerd en Data-Becker doet er uitstekende zaken mee. Over hun premiereboek over de 128 zijn we niet zo enthousiast, men ramt dat soort werkjes in elkaar, terwijl de machine zelf in zijn definitieve vorm nog niet eens beschikbaar is. Berlijn was wat betreft Commodore voor ons een beetje een teleurstelling, men is daar de markt een beetje aan het verliezen aan Amstrad en misschien MSX en een wat hautaine nadruk op de busines-machines wordt ze door de C-64 eigenaars niet in dank afgenomen.

Londen

Op de PCW Show in Olympia in Londen was dat iets rooskleuriger, daar kon men de C128 in vele variaties bewonderen, al waren het nog steeds prototypes. Met veel belangstelling van het publiek en de soft-

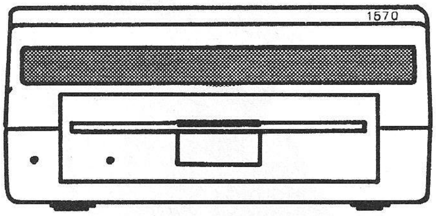
ware-huizen blijven met veel tam-tam hun produkten pushen. De afgang van Sinclair - en de kater voor de winkeliers die met hun voorraden blijven zitten - laat bovendien nogal wat ruimte in de markt en zowel winkeliers als software-ontwikkelaars moeten dus wel en zijn ook redelijk optimistisch over het komende seizoen. Betere actiespelletjes, een ware golf adventures (vaak met geldprijzen voor de eerste oplosser), goedkopere periferie en het begin van het schoolseizoen werken samen om de moed er in te houden. Wel moet ook in Engeland de 128 de grote redding worden en het lijkt wel, of de sprong naar de 128 KB epidemisch is. Ook Amstrad heeft nu een 128 KB model, Atari heeft de 130 XE en ook de MSX machines krijgen een groter geheugen in de MSX-2 versie.

Parijs

In Frankrijk is Commodore absoluut niet het belangrijkste huiscomputermerk, bedrijven als Apple, Thomson, Matra en ook Amstrad en zelfs Philips vinden daar meer weerklank. Toch was er op de SICOB tentoonstelling een grote CBM stand met een twintigtal CBM-128's, ook weer prototypes. En natuurlijk de 1571, wat PC-10's en zelfs een nieuw combinatiepakket met de Plus-4. De Fransen hebben het niet makkelijk, de aansluiting aan de TV (SECAM-standaard) moet altijd via speciale aanpassingen voor de PAL standaard lopen, dat vertraagt de komst van machines. Ook speciale Azerty toetsenborden met alle accenten zijn nodig en die had men bv. op de 128's al aangebracht. Er was erg veel belangstelling bij het publiek voor de machines, het zag er werkelijk zwart van de mensen. Temidden van die drukte ontdekten wij toch nog een Commodore wereldprimeur, namelijk een hele nieuwe diskdrive.

1570

Wie aan een 128 denkt, krijgt al snel de indruk, dat hij dan ook maar snel een 1571 moet aanschaffen, om ten volle alle mogelijkheden van die machine en met name de CP/M optie te kunnen gebruiken. Dat is niet helemaal waar, de 1541 werkt in principe in alle drie de modes samen met de 128. Dat is in tegenstelling tot eerdere berichten, waarin werd gesteld dat er voor de CP/M mode altijd een 1571 nodig zou zijn. De 1541 kan wel degelijk onder CP/M werken, maar dan alleen met speciaal geformatteerde schijven, die volgens de Commodore manier zijn beschreven. De 1541 kan geen standaard CP/M schijven van bv. de Osborne of Kaypro lezen. Een nadeel van de 1541 blijft zijn beperkte capaciteit en de snelheid, die berucht langzaam is en in de CP/M mode wel heel traag.



Voor wie toch meer snelheid wil hebben, ook normale CP/M schijven wil kunnen lezen en toch niet zoveel wil uitgeven, heeft CBM nu de 1570 gemaakt. In hetzelfde kastje als de 1541 en gelijk aan de 1571, alleen met een enkelzijdige drive in plaats van een dubbelzijdige. Speciaal ontwikkeld voor de Europese markt, en over deze drive leest u dus nu voor het eerst in dit blad, in Amerika weten ze nog van niets. Een goedkopere versie van de 1571, zonder het mooie kastje en met wat minder capaciteit, maar heel aantrekkelijk voor wie toch een "echte" 128 installatie op het oog heeft. ●

nieuws

Video-digitizer voor Commodore

De Amerikaanse firma Cardco introduceert het Digi-Camsysteem bestaande uit een Panasonic monochrome (=zwart-wit) videocamera, digitalisator, bijbehorende software en kabels. De Cardco digitalisator zet de door de still-videocamera opgenomen informatie in 3 seconden op het tv/monitorscherm. Het zwart/witbeeld bestaat uit 320-200 pixels in vijf grijstrappen.

Alle opgenomen beelden kunt u via de printafdrukken, op schijf vastleggen of via een telefoonmodem versturen. De software maakt het mogelijk de videobeelden op eigen smaak af te stemmen.

Adres: Cardco Inc.,300 South Topeka, Witchita, KS 67202, USA.

Doe het zelf PC

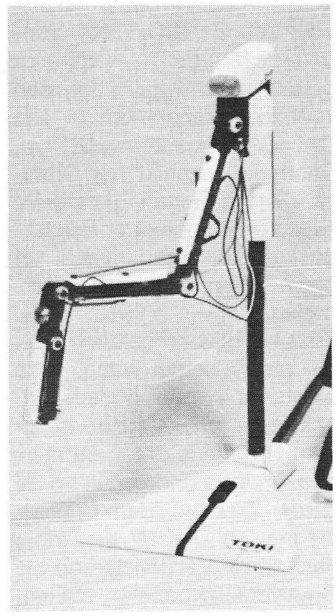
Voor de thuiscomputeraar met pretenties bleef een echte IBM-PC-compatibele computer door de stevige prijs tot op heden een wensdroom. Nu niet meer, want de Duitse Firma AMTEC uit Giessen brengt een complete IBM-PC bouwkit op de markt . Deze kit bestaat uit een moederkaart met 64KB RAM, alle benodigde prints, toétsenbord en een bijpassende behuizing. Uitbreiding naar eigen keuze is mogelijk en het zelf construeren heeft leuke educatieve kanten voor hen die altijd al eens in het PC-binnenwerk wilden rondsnuffelen.

Wolters Assembler

De machinetaalfans, of eigenlijk degenen die dat willen worden, kunnen nu bij Wolters (dus via V&D, MCN en de boekhandel cq. computershops) een uitgebreide cursus **In Assembler programmeren op de 64** aanschaffen. Als gebruikelijk bij Wolters weer in een van die supergrote verpakkingen, maar het A-4 formaat boek telt dan ook bijna 200 pagina's. Er zijn twee cassettes bij. Op een daarvan staat een Assembler, dus een programma om in Assembly te kunnen programmeren. We bekeken die assembler en het is als leermiddel zeer goed te gebruiken, voor het maken van uitgebreide machinetaal-programma's moet men later wel een echte assembler gaan aanschaffen met betere edit mogelijkheden. Maar als cursushulpmiddel is het programma goed te gebruiken. Samen met het duidelijke en goed verzorgde cursusboek is deze cursus voor *f* 129,- een aantrekkelijk aanbod.

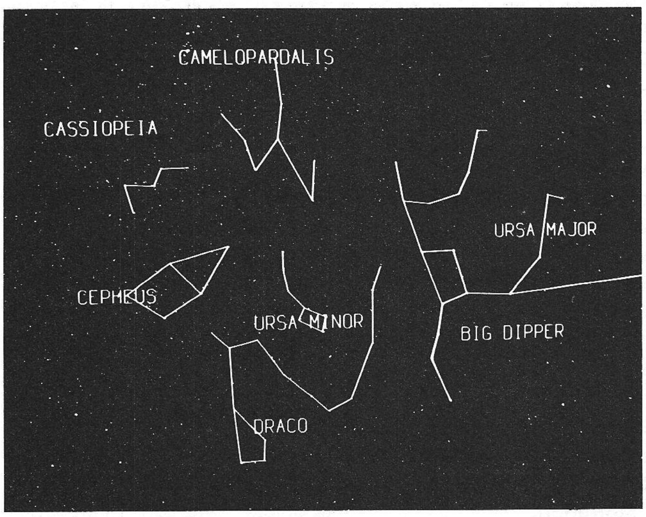
Biometaal

Het biometaal of biowire van Toki Corporation Japan is een speciale titanium-nikkellegering die onder invloed van een elektrisch stroompje van vorm kan veranderen. Wordt deze legering in een spoelvorm gefabriceerd dan kan het net als



een menselijke/dierlijke spier samentrekken en weer naar de uitgangspositie terugkeren. Stroom aan, het spoeltje verkort zich. Stroom uit, het spoeltje veert terug naar de "geheugenstand". Biowire leent zich daardoor uitstekend voor de fabricage van robots. Toki zelf levert al een kleine Biometalarm.

NIEUWS NIEUWS NIEUWS



Digistar

Het supergrootbeeld Omniversumtheater in Den Haag heeft er een nieuwe attractie bij en daarbij speelt de computer een hoofdrol. De DIGISTAR computergestuurde projector bouwt punt voor punt een kompleet planetarium of een CAD/CAM-lijnentekening op. Daarmee beschikt het Omniversum als enige in de wereld over een kompleet computergestuurd planetarium met tot op heden ongekende mogelijkheden.
Zo is het bijvoorbeeld mogelijk de sterrenhemel vanaf elke plaats (ook vanaf andere planeten) en in elke tijd te bekijken. Op Venus draait de sterrenhemel de andere(als op de aarde) kant op. Idem dito als wij met de Digistar een tripje naar deze Morgenster op de Omniversumkoepel maken. Behalve interplanetaire "reizen" biedt de Digistar ook interstellaire "excursies". Willen wij weten hoe de hemel er in een sterrenstelsel op 1500 lichtjaren afstand van onze aarde uitziet, dan zoeft de Digistar in 15 seconden door het universum naar de doelsector, terwijl alle gepasseerde hemellichamen in de exacte helderheid worden weergegeven.
Momenteel worden er slechts enkele mogelijkheden van de Digistar benut. De krachtige VAX 11/370 computer van Digital Equipment Corporation is tot vele verrassende CAD/CAM-effecten in staat. Tot de verbeelding spreken natuurlijk Star Wars-achtige ruimteslagen, het ontwerpen van futuristische modellen en het simuleren van het ontstaan der sterren uit galactische gaswolken.
De Digistar heeft echter ook nog praktische mogelijkheden: aangezicht, diepte en (onderlinge-) verhoudingen zijn van elk willekeurig voorwerp, model, voer- en vliegruimtetuig en gebouw in het geheugen op te slaan. Vervolgens kan men op de Omniversumkoepel een geprojecteerd model van alle zijden bekijken, door de straten van steden lopen en daar zelfs nieuwe bouwwerken in projecteren om te kijken hoe deze stedebouwkundig uitkomen. Eventueel zou men zelfs de inpoldering van de Waddenzee met alle waterloopkundige gevolgen van dien kunnen visualiseren! En dat alles voor 300 toeschouwers tegelijk.
Bel voor reservering en nadere informatie het Omniversum: tel.070-545454.

Een eigen tekenset maken, dat is niet alleen handig om misschien weer eens wat andere letters op het scherm te krijgen, het is ook de basis voor een veel leuker arsenaal symbolen. Daarmee vallen bijvoorbeel hele leuke achtergronden te maken voor een spel. Martin Piller en Peter van der Toorn over het zelf programmeren van een tekenset en standaardfiguurtjes.

Programmeerbare karakters ontwerpen



en alles wat er mee samenhangt

Het maken van een goed videospel berust niet alleen op het gebruik van sprites, zoals we die in eerdere artikelen behandelden. Men zal naast de sprites toch ook een fraaie achtergrond willen kreëren of aardige additionele vormen of figuurtjes willen gebruiken.

Als eerste stap op weg naar een volledig 'gebitmapped' scherm, waarbij ieder beeldpunt afzonderlijk te plaatsen is - dus 360 x 200 beeldpunten -, zullen we het programmeren van eigen karakters behandelen. Wat we op het beeld zien aan letters en andere tekens, kunnen we veranderen in figuren of delen van figuren. Ook voor deze grafische grappen zullen we gedwongen zijn ons te bedienen van POKE's en PEEK's. De Commodore 64 zal standaard nooit de mogelijkheid bieden eenvoudig tekeningen en plaatjes, gedetailleerder dan een uit verschillende karakters van de tekenset bestaande illustratie, te maken. De weg via POKEs en PEEKs biedt daarentegen meer mogelijkheden op grafisch gebied en is eenvoudig te leren.

Tekens of karakters

In het kader van dit onderwerp zullen wij de volgende zaken behandelen, waarbij we de gewenste symbolen voor het gemak maar karakters zullen, noemen. Wel wat Engels, maar wie veel met dit soort computerjargon werkt, neemt dat toch gauw over.

- Waaruit bestaat een karakter; hoe is het opgebouwd?
- Waar leest de computer de inhoud van de karakters?
- Waar kunnen wij de nieuwe inhoud van de door ons ontworpen tekens plaatsen (in het geheugen) en hoe vertellen we de computer dat hij ze daar moet gaan uitlezen?
- Welke toepassingen zijn er op dit gebied mogelijk?

De opbouw van een karakter.

De karakterset (dat zijn alle lettertjes, cijfertjes en vormpjes) waarover de Commodore 64 en overigens alle andere CBM- computers beschikken, bestaat uit 512 verschillende tekens. De karakterset omvat vier groepen; te weten hoofdletters en kleine letters die op hun beurt diapositief en dianegatief uitgevoerd kunnen worden. Iedere groep bestaat dus uit 128 tekens.
Wij kunnen echter slechts over 256 tekens tegelijkertijd beschikken. Er is een keuze te maken uit een van deze beide groepen (dat kan met SHIFT + C=. De VIC-20 bezit overigens de mogelijkheid om wel over alle 512 karakters tegelijk te beschikken; door in het adres 36867 bit 0 aan te zetten oftewel er een oneven waarde in te POKEn (POKE 36867, PEEK(36867) OR 1), wordt het beeldscherm twee keer zo hoog! De tekst is nu niet meer leesbaar; dat komt doordat de videochip in de zogeheten 8 bij 16 mode staat.

Keren we nu terug naar de Commodore 64.

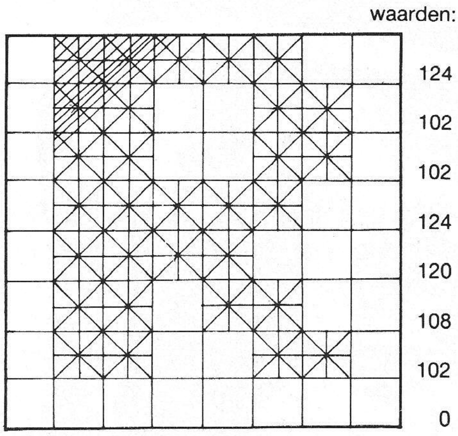
Hier is elk karakter opgebouwd uit een matrix van 8 bij 8 beeldpunten. Een horizontale reeks van 8 beeldpunten is een byte, een karakter is dus uit groepje van acht bytes die onder elkaar liggen opgebouwd. Elk groepje apart is genummerd, zo heeft de A de waarde 1, de B de waarde 2 enz. Met andere woorden: bij elk karakter uit de tekenset hoort een getal dat tussen de 0 en 255 ligt; raadpleeg hiervoor de tabellen in het handboekje (bijna achterin). LET OP: dit zijn niet de CHR$-codes, maar de beeldschermcodes.

Print of Poke

Er zijn twee manieren om een karakter op het scherm te krijgen. Ten eerste de alom bekende PRINT-instructie welke niet veel uitleg behoeft te krijgen. Een tweede methode is die via de POKE- instructie. Deze instructie maakt het mogelijk om in een bepaalde byte ergens in het geheugen, een waarde te zetten. Willen we nu een karakter op het scherm zetten dan kan dat door een bepaalde waarde in het zogeheten beeldschermgeheugen te POKEn. Het beeldschermgeheugen is 1000 bytes groot en start normalitair op adres 1024 (het is mogelijk dit startadres te wijzigen). Als we nu bijvoorbeeld een 'A' linksbovenaan op het scherm willen zetten, dan voeren we in: POKE 1024,1 (return).

Dit kan natuurlijk ook in programmavorm. Bij deze instructie stelt het getal 1024 de plaats van het karakter voor en de 1 welk karakter. De waarden die bij de verschillende letters/karakters horen kunt we vinden in de 'POKE'-waarden tabel in de, normaal met de computer meegeleverde, 'USER's GUIDE' of handboek.

Het is dus mogelijk om door middel van het vullen van 1 (beeldscherm)geheugenlocatie een groepje van 8 bytes op het scherm te krijgen! Laten wij nu als voorbeeld de letter 'R' nemen en kijken hoe deze is opgebouwd uit dat groepje van acht bytes.



△ figuur 1.

De inhoud van 1 karakter wordt dus gedefinieerd door 8 bytes. Aangezien er 512 verschillende karakters aanwezig zijn, kost het de computer 512x8=4096 bytes geheugen (dat is 4K) om alle karakters te definieren. Deze 4096 bytes worden wel de KARAKTER GENERATOR ROM genoemd. ROM houdt in dat we in dit geheugen niet kunnen schrijven maar er alleen uit kunnen lezen. Het is dus normaal gesproken NIET mogelijk om de vormgeving van de karakters te veranderen. De karakter generator begint op het adres 53248 en loopt tot 57343. In dit gebied zitten echter tegelijkertijd allerlei Input/Output aangelegenheden zoals de video-chip en de geluids-chip. We moeten dus een keuze maken: of we laten dit gebied fungeren als I/O gebied of als karakter generator. Deze keuze wordt bepaald door byte 1; indien bit 2 van deze byte aan staat (de normale situatie) dan fungeert dit gebied als I/O gebied. Zetten we deze bit uit dan wordt dit gebied karakter generator!
We mogen bit 2 van byte 1 overigens niet zomaar uitzetten; we moeten dan eerst het I/O gebied stilzetten want anders gaat er een aantal dingen mis. Het I/O gebied zetten we stil door in byte 56334 bit 0 uit te zetten en maken we weer actief door het bit weer aan te zetten. Samengevat:

Normaal gesproken is het gebied 53248 tot 57343 een I/O gebied. Door nu eerst in byte 56334 bit 0 uit te zetten en vervolgens in byte 1 bit 2 uit te zetten verandert dit gebied in de karakter generator; hierin is de vormgeving van alle 512 karakters vastgelegd.

Geheugenplaats

Hoe kunnen wij de computer vertellen waar een en ander zich bevindt?
De plaats (in het geheugen) waar de computer de inhoud van de karakters leest, hoeft niet per se de karakter generator te zijn; we kunnen namelijk deze plaats zelf kiezen. Op die manier is het mogelijk om de computer de inhoud van de karakters in een stuk RAM te laten uitlezen. RAM is geheugen waarvan we de inhoud zelf kunnen bepalen.

Bit 1, 2 en 3 van het register 53272 bepalen de plaats waar de computer de karakters uitleest; de volgende tabel geeft aan hoe wij bit 1, 2 en 3 moeten kiezen om het STARTadres van deze plaats te bepalen:

Bit-plaats

Figuur 2 toont een tabel waarin de stand van de bits staat die moet worden gehanteerd om de computer zijn karakters in een bepaald gebied te laten lezen. De videochip bezorgt ons een complicatie waarwe goed van op de hoogte moeten zijn. Hij heeft namelijk de eigenschap dat hij maar 16K van de 64K bytes van het geheugen tegelijk kan 'zien'. Wat wordt daarmee bedoeld? De videochip heeft naast de taak de karakters in de juiste vorm uit te lezen nog meerdere funkties. Hij moet de sprites besturen, de karakters

op het scherm plaatsen ed. Wellicht was het al opgevallen dat bij het programmeren van sprites de data van de sprites slechts in de eerste 16K van het geheugen worden geplaatst. De sprite data pointer bestond immers uit slechts een byte, waarvan de inhoud met 64 vermenigvuldigd moest worden om de juiste plaats te vinden. Nu is het normaal gesproken zo dat de videochip slechts de eerste 16K bytes gebruikt om zijn informatie op te slaan; in deze 16K staan dus de sprite data pointer met de data van de sprite, het beeldscherm geheugen wat namelijk in het 1000 bytes grote gebied na adres 1024 zit en de karakter-vorm informatie. Een slimme jongen zal nu zeggen dat de normale vorm van de karakters toch 'achter' de videochip in het gebied vanaf adres 53248 zit. Dit is zo, maar de videochip wordt wijs gemaakt dat het niet daar, maar in een gebied in de eerste 16K te vinden is.

Hiermee zijn we dan weer aangeland bij de tabel van figuur 2. De tabel toont zoals gezegd de bitpatronen van bit 1,2 en 3 in byte 53272. Achter ieder bitpatroon staat een getal het startadres van de karakter generator voorstelt. De waarden XXXX010X en XXXX011X corresponderen normaal gesproken met het gebied na 4096 en 6144, maar op dit punt wordt de videochip beduveld; wanneer deze waarden in het adres 53272 staan dan leest de computer in het ROM-gebied en niet in de desbetreffende gebied uit de tabel!

Plaats

Zoals gezegd kan de plaats van het beeldschermgeheugen zelf worden bepaald. Wat bedoelen we daarmee? De computer kijkt telkens welke tekens/karakters weergegeven moeten worden op het scherm. Daartoe leest hij steeds een bepaald geheugengebied uit dat het beeldschermgeheugen wordt genoemd. Al eerder zagen we hoe we, door een waarde in dit geheugengebied te POKEn, een karakter op het scherm konden plaatsen. Dit geheugengebied zelf kunnen we van plaats veranderen, ook dit geschiedt in 53272 (nu gebruiken we dus de eerste 4 bits). Overigens is een nybble een groepje van 4 bits; een byte bestaat dus uit 2 nybbles.

Nu zijn we er nog niet; we hebben nu alleen de video-chip verteld waar het beeldschermgeheugen zit, maar de KERNAL (het operating system dat bv. steeds READY vermeldt) weet nog van niks. De KERNAL stuurt zijn beeldschermoutput nog steeds naar het oude beeldschermgeheugen (dus naar 1024 en verder). Nu gebruikt de KERNAL een byte waarin het startadres van het beeldschermgeheugen is vastgelegd. Dit doet byte 648; de inhoud van deze byte vermenigvuldigd met 256 levert het startadres van het beeldschermgeheugen op. Het spreekt vanzelf dat de waarde die in 648 wordt gePOKEd overeen moet komen met de waarde die we aan de video-chip hadden doorgegeven, (maar dat had U zelf natuurlijk al

0	0	1	1	1	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	0	1	0	0	1	0	0
0	0	0	1	1	0	0	0
0	1	1	0	0	1	1	0
0	0	0	0	0	0	0	0

door!). Bevindt het beeldscherm zich op bijvoorbeeld 1024 (de normale situatie), dan hoort in byte 648 de waarde 1024/256 = 4 te staan. Het kleurengeheugen, dat belangrijk is omdat via het POKEn van karakters op het beeldscherm anders niet de mogelijkheid geboden wordt deze karakters in verschillende kleuren uit te voeren, kan NIET worden verplaatst! Het start op 55296 en is (uiteraard) 1K groot, (waarom?). Als we nu een karakter op het beeldscherm hebben gePOKEd, maar het is niet zichtbaar, dan moeten we de karakterkleur (nog) wijzigen.

Generator

Als we de tabellen van fig. 3 beschouwen, zien we dat het varieren van de beeldschermplaats/karakter-generator steeds beperkt is tot 16K. Dat komt omdat de videochip steeds een gebied van 16K gebruikt, waarin hij het beeldschermgeheugen/karakter generator uitleest. Willen we nu een ander gebied van 16K gaan gebruiken, dan kan dat ook weer. Er is namelijk een zogenaamde BANK. Deze bepaalt welk gebied (van 16K) door de videochip wordt uitgelezen. Er zijn dus 4 plaatsen in het geheugen aanwijsbaar die als begin van zo'n 16K gebied kunnen fungeren. We maken nu gebruik van register 56576 en daarvan de eerste 2 bits, zie fig. 4. De plaatsen van dat gebied van 16K wordt gevonden door de gezamelijke waarden van deze 2 bits met de factor 16384 te vermenigvuldigen. De plaats van (het startadres) van beeldschermgeheugen/karakter-generator is nu de beginwaarde van de BANK plus de waarde uit de desbetreffende tabel. Het is dus niet mogelijk om bijvoorbeeld het beeldscherm in 1024 te kiezen en tegelijkertijd de karakter-generator in 32768 te hebben; deze moeten beide in dezelfde BANK liggen!

Commodore/shift

Nu U veel kundiger bent geworden vraagt U zich natuurlijk direct af hoe de SHIFT-COMMODORE mogelijkheid (C=) werkt. Het in indrukken van de beide toetsen doet de computer overschakelen van hoofdletters op kleine letters en omgekeerd. Wat er nu daadwerkelijk gebeurt, is niet veel meer dan dat de computer overschakelt op een 'andere' karakter-generator. Staan de waarden voor de normale hoofdletters in een bepaald gebied dan vinden we de vormen voor de kleine letters een 2048-tal bytes verderop.

De praktische toepassingen.

Het lijkt ons het beste dat we dit aan de hand van een voorbeeldprogramma bespreken.

Zie hiervoor het onderstaande BASIC-programma.

```
5 POKE53280,13: POKE53281,13
10 PRINTCHR$(144)
20 POKE56334, PEEK(56334) AND
   254
30 POKE1, PEEK(1) AND 251
40 FORT=0TO4095
50 POKE8192+T, PEEK(53248+T)
60 NEXT T
70 POKE1,PEEK(1)OR4
80 POKE56334, PEEK(56334) OR 1
```

```
90 TEKEN=97:PLAATS= 8192+
    TEKEN*8
100 REM ........................
110 REM ........................
120 REM ........................
130 REM ........................
140 REM ........********........
150 REM .....**************.....
160 REM ...******************...
170 REM ..***..***....***..***..
180 REM .*****..*********..*****.
190 REM .*****..*********..*****.
200 REM .......**********.......
210 REM ......*****..*****......
220 REM .....****......****.....
230 REM .....***........***.....
240 REM ....****........****....
250 REM ....*****......*****....
260 REM ....*******..*******....
270 REM ....****************....
280 REM ....****************....
290 REM ........................
300 REM ........................
310 REM ........................
320 REM ........................
330 REM ........................
400 FORT=0TO71:READ P:POKE
    PLAATS+T,P:NEXT
410 DATA 0,0,0,0,0,7,31,57
420 DATA 0,0,0,0,255,255,255,199
430 DATA 0,0,0,0,0,224,248,156
440 DATA 124,124,1,3,7,7,15,15
450 DATA 255,255,255,231,
    129,0,0,129
460 DATA 62,62,128,192,224,
    224,240,240
470 DATA 15,15,15,0,0,0,0,0
480 DATA 231,255,255,0,0,0,0,0
490 DATA 240,240,240,0,0,0,0,0
500 POKE 53272,(PEEK(53272) AND
    241) OR 8
510 PRINTCHR$(147)+CHR$(17)
520 A$=CHR$(161)+CHR$(162)
    +CHR$(163)
530 FORT=1TO3:
    A$=A$+CHR$(157): NEXT:
    A$=A$+CHR$(17)
540 A$=A$+CHR$(164)+
    CHR$(165)+CHR$(166)
550 FOR T=1TO3:
    A$=A$+CHR$(157): NEXT :
    A$=A$+CHR$(17)
560 A$=A$+CHR$(167)+
    CHR$(168)+CHR$(169)
570 A$=A$+CHR$(145)+CHR$(145)
580 FOR T=0TO12: PRINT A$;:NEXT
590 PRINT:PRINT: PRINT: GOTO580
```

Een aantal opmerkingen ter verduidelijking

Regel 20: De I/O wordt 'stilgezet'; de RESTORE toets werkt nu ook niet meer!
Regel 30: Omschakelen naar tekengenerator
Regel 40-60 Inhoud karakter-generator ROM gekopieerd in RAM; onze nieuwe karakter-generator. Dit kan enige tijd vergen.
Regel 90 Nieuwe tekenset definiëren op de plaats van het normale karakter uit de tekenset, met de POKE-waarde die staat in de variabele TEKEN.
De vorm van een is vastgelegd als punten en sterren binnen de REM-opdrachten van regel 100 tot 330. Het is een groepje van 3 karakters breed en 3 lang. Nadat we ze omgerekend hebben naar een waarde tussen 0 en 255, hebben we deze waarden, 8 per karakter, geplaatst in de data regels 410 tot 490.
Regel 400: Inlezen tekens van l naar r en van boven naar beneden.
Regel 500: De computer gebruikt nu het door ons gekozen gebied na 8192 als karakter-generator. De letters blijven ongeschonden omdat wij voordat we hierop overschakelden de inhoud reeds kopieerden.
Regel 520-570 De vorm van een telefoon hebben wij in een (A)-string gezet om deze als geheel op het scherm te plaatsen. Duidelijk is dat hoewel we steeds rekenden met de POKE-waarden, we de karakters die we in de string zetten dus ook met een CHR$-waarden of het toetsenbord kunnen aanroepen; we zijn dus niet gebonden aan het op het beeldscherm POKEn van de karakters. De manier waarop de (A)- string is samengesteld wordt duidelijk als de CHR$-waarden tabel wordt beschouwd. Vergelijk POKE-waarde 97 met CHR$-waarden 161.

Het besproken voorbeeldprogramma heeft niet bepaald nuttige toepassingen, edoch wij verstrekten dit teneinde de eerder uitgediepte droge stof in een wat luchtiger kader te plaatsen.
In spelletjes kunnen achtergrond of stilstaande voorwerpen worden getekend, bij bepaalde onderwijs- of simulatieprogramma's kunnen bepaalde niet tot de standaardtekenset behorende tekens, zelf worden gecreerd, maar ook indien men de standaard karakterset niet mooi vindt, is het maken van een eigen lettertype (spijkerschrift e.d.) een uitkomst. ●

Gebruikersgroepen

De **VCGN** (070-971851) is de grootste Commodore gebruikersvereninging en zit iedere laatste zaterdag van de maand in het Lodewijk Makeblijde lyceum in Rijswijk.
Inlichtingen over de vele afdelingen kunt u krijgen via:
Zeeland 01180-25746 West-Brabant 01640-51639 Midden-Brabant 01620-55634 Oost-Brabant 04120-42054 Zuid-Limburg 043-476935 Zuid-Holland 010-501023

PET Benelux heeft bijeenkomsten in Zeist (1e zaterdag), Nijmegen (2e zaterdag) en Haaksbergen (3e zaterdag). (Inl.05759-2211).

De **HCC** Commodore gebruikersgroep heet Compet, 03404-59599.

HOT NEWS van SCN uit Amsterdam (03450-16051) heeft haar grote bijeenkomst in de Meervaart op de 2e zaterdag van de maand en ook activiteiten in Zwolle (038-539760), Midden Nederland (03453-1375) en Deventer (05700-50758).

WZW is een van de clubs uit Belgie, met afdelingen in Antwerpen, Limburg, West-Vlaanderen inl. Bindstr. 19, te Berchem Antwerpen.

Ook in Gent is er een groep, te bereiken via 091-253937

In Arnhem zit ook een Commodore-club, waar men clubavonden op dinsdag iedere twee weken organiseert en ook cursussen geeft. Inl. Talingsingel 32 te Velp, tel. 085-647782.

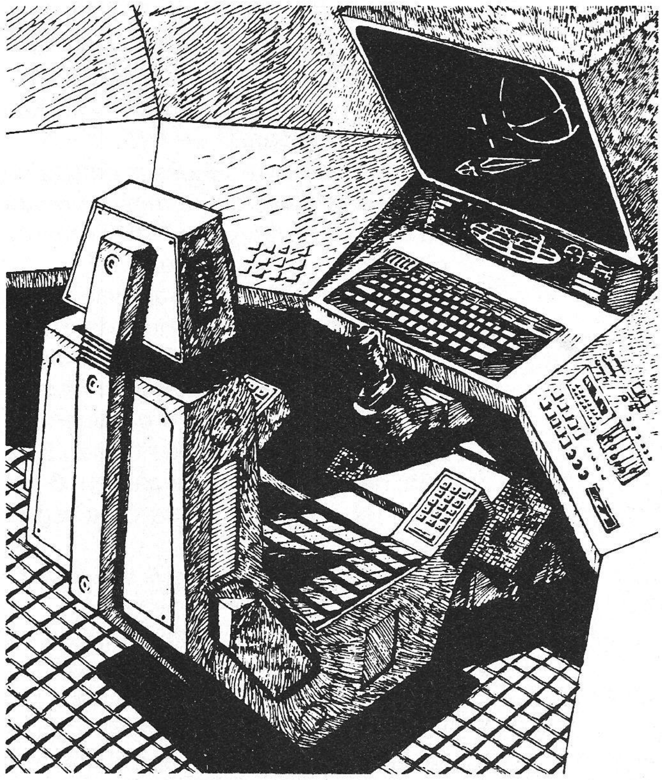
In het Belgische **Knokke-Heist** is ook een gebruikersclub actief. Men is daar heel ruimdenkend, de bijeenkomsten staan open voor gebruikers van alle merken home computers. Elke vrijdag vanaf 20.00 uur komen de computerenthousiasten uit de omgeving bijeen in Café Monty in Knokke. Op zaterdag komt men bij elkaar in de Broederschool in Duinbergen van 9.30 tot 12.00 uur. Het contactadres is: Computerclub Knokke Heist, p.a. Bob Dyserink, Kerkstraat 19, 8390 Knokke Heist, België.

Elite

Van Firebird via Micromundo

In Engeland een topper en er zijn in de versies voor de Acorns al meer dan 100.000 verkocht. Ian Bell en David Braben ontwierpen deze combinatie van strategisch ruimtemonopoly en schietspel, waarvan we de originele, complexe en goed doordachte structuur eigenlijk qua grafische uitvoering vinden tegenvallen en de prijs van meer dan 90 gulden aan de hoge kant is.

Het is maar heel zelden, dat bij een spel een echt goede documentatie wordt geleverd. Bij Elite is dat het geval, en nog wel in het Nederlands en daarmee verdient importeur Micromundot een compliment. Die Nederlandse handleiding is voor een deel ook de verklaring voor de prijs.
De verschillende onderdelen, zoals de hele vlieg en navigatieprocedure, de interplanetaire reizen met de strijdvoering van het vaartuig, de Cobra MKIII, de handelsregels, een politiek profiel van het heelal en een lijst van vijandelijke schepen met beschrijving, het lezen van het boekje houdt je ook nog wel een paar uur bezig. Er is een verbazingwekkende hoop ideëen in gestopt, Elite is een compleet bedacht SF scenario en we vragen ons af, wat een werkelijk goede grafische uitvoering op bv. de Amiga van dit spel zou maken.



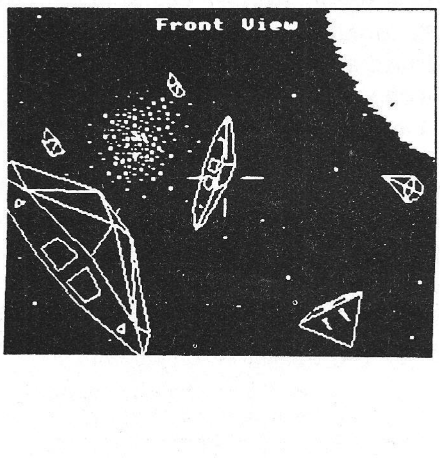
Een blik op de brug van Cobra MKIII toont duidelijk het scannerscherm, de astrogatie- console en de monitor van het hoofdsysteem.

Vector beweging

We vinden het persoonlijk wat mager, maar de uit lijnen opgebouwde spelbeelden bieden wel een vrij unieke driedimensionale ruimte om in te manoevreren. Misschien zijn we verwend door wat grotere computers op dit gebied kunnen, want op zich is dit op een C-64 nog niet vertoond. De computer is driftig aan het werk om de bewegingen en draaïngen mogelijk te maken en het resultaat zijn van die opengewerkte vormen, die echt driedimensionaal meedraaien.
Als je gewend bent aan de vectorbeelden, komt overigens de strategische waarde van het spel pas tot zijn recht en ga je ook de instrumenten onderin beeld beter benutten. Het geluid is vrij mager en op den duur wat irritant, een beetje intelligente spraaksynthese was meegenomen geweest.

Zaken doen

Het leuke van Elite is dat er naast snel reageren ook wel degelijk nagedacht moet worden, wil men tenminste wat verdienen met de interstellaire (ruimte) handel. En dat is weer nodig om het ruimteschip, dat je bestiert, uit te rusten met de nodige wapenrusting. Elite is als spel veel meer dan een kortstondig testen van de reactiesnelheid, het gaat er ook en juist om door slim handel te drijven en je reis langs allerlei sterren goed uit te stippelen. Handelen in goederen als voedsel, kernbrandstof, slaven (?), drugs, computers en metalen, die per planeet verhandeld kunnen worden, en natuurlijk zijn de prijzen voor agrarische spullen op een technische planeet hoog en omgekeerd. Maar wie niet wil handelen, kan ook gaan bonusjagen of piraat spelen, of een asteroide ontginnen of zelfs smokkelen.
Wie slaagt met dit spel bereikt de Elitestatus, waar het spel naar genoemd is, maar trek daar gerust een paar weken voor uit! Maar wie nog langer wil doorgaan, met 8 melkwegstelsels en een paar duizend planeten om te exploreren kun je nog wel even voort.



Mercenary

van Novagen via Topsoft

In dezelfde lijn als Elite, maar zonder de Nederlandse handleiding en dat maakt dit spel wel wat goedkoper, het gaat hier 49 gulden kosten. Qua driedimensionaal grafisch werk en vliegbewegingen een kopie van Elite, al is hier het verhaal natuurlijk wel anders.

Dit spel is gemaakt door Paul Woakes, die eerder Encounter! en het Novaload snellaadprogramma ontwikkelde en werd op de PCW show in London voor het eerst getoond. De basis van het scenarion is een ontsnapping van de planeet Targ, waar een denkbeeldige stad/wereld is gevestigd, bestaande uit wegen en allerlei gebouwen en structuren, die zich deels onderaards uitstrekken. Met de hulp van Benson, zijn persoonlijke vechtcomputer, moet de speler, die een soort huursoldaat is, de planeet verkennen en vijanden verslaan. Dat kan in verschillende voertuigen en die 3-D tripjes zijn daarom erg gevarieerd. Maar het is uiteindelijk om rijkdom te

doen, dus de vrienden en vijanden kunnen nogal eens wisselen. Tenslotte moet de vergaarde rijkdom voldoende zijn om van de planeet weg te kunnen, maar om zover te komen zijn er eindeloze mogelijkheden. Gezien de vele speelvarianten is er een zeer welkome "Save gameplay" funktie, waarmee een bepaalde situatie kan worden vastgelegd om later door te spelen.

Vervolgserie

Het is de bedoeling, dat er in totaal drie delen van de Mercenary reeks gaan komen, waarvan dit de eerste is. Wie uit Targ weet te ontsnappen, heeft meteen de sleutel om in Mercenary II door te kunnen gaan.

Epyx

Vanuit Amerika kregen we bericht, dat het toonaangevende sofwarehuis Epyx in Europa nieuwe wegen heeft gevonden om haar produkten op exklusieve basis te distribueren. In Engeland wordt dat US GOLD en voor de Benelux gaat het via Editions & Marketing Multi Media lopen.

Bat Attack

Alligata Software

Een leuk en goedkoop spelletje. De maker heeft twee mogelijkheden ingebouwd: of je speelt d.m.v. het toetsenbord of met een joystick. Dat laatste is aan te bevelen, want al dat geram op het toetsenbord kan je wel eens opbreken, een nieuw toetsenbord is veel duurder dan een joystick ! In dit spel heb je op de onderste beeldrij de beschikking over een soort afweergeschut. Met de vuurknop of met de Shift-toets kun je schieten. Daarnaast kun je het geschut snel van links naar rechts verplaatsen en andersom. Je wordt aangevallen door een soort vleermuizen die ook nog bommetjes kunnen afgooien. Wordt je getroffen door een vleermuis of een bom, dan ben je een stuk geschut kwijt. Ben je drie keer getroffen dan is het "game over". Het leuke van dit spel is dat wanneer jij je geschut verplaatst de vleermuizen direkt van richting veranderen en op je af komen. Heb je alle vleermuizen uitgeschakeld, dan wordt je bestookt met een soort schijven. Deze moet je zien te ontwijken. Lukt dat, dan komt de volgende ronde, die weer moeilijker is. Bij elke 10.000 punten krijg je er een stuk geschut bij. De hoogste score wordt door het programma bij gehouden.

Assembler 64

Interceptor Software

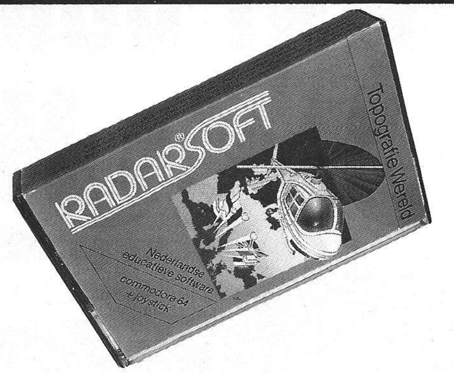
Verwacht er niet te veel van, al is het vrij goedkoop. Toch is dit echt wel een leuk programma voor hen die pas beginnen of gaan beginnen op het gebied van machinetaal.
Het programma is zo gemaakt, dat je net als in BASIC gebruik maakt van regelnummers. Dit is erg handig, omdat je, indien er wat fout zit, makkelijk regels kunt toevoegen. Dit is bij vele andere ASSEMBLERS niet zo eenvoudig. Bij een sprongopdracht kun je gewoon verwijzen naar het betreffende regelnummer. Het programma stelt je bovendien in staat te werken met variabelen. Je benoemt bijvoorbeeld de startpositie van je scherm (1024) als SC. Heb je later in het programma SC nodig, dan kun je er gewoon mee "rekenen". Bijvoorbeeld: X=SC+5. Het programma is voorzien van een laad en save routine, zodat je alles netjes op tape kwijt kunt.

Topografie Nederland

Radarsoft

Al sinds dit voorjaar op de markt, wie kent dit programma nog niet. Het heeft veel aandacht en publiciteit gehad en terecht, we vinden dit eigenlijk wel het "Nederlandse programma van het jaar". Zeker ook, omdat het de voorloper is van een hele internationale reeks en ook een exportsucces voor Nederlandse software.

Met dit programma kun je spelenderwijs je topografische kennis ophalen. Je kunt het programma vragen om een bepaalde plaats of rivier op te zoeken, maar je kunt ook een "stedenspel" spelen. De bedoeling is dan om binnen een bepaalde tijd 15 steden of eilanden met een helicopter te bezoeken. Lukt je dat, dan krijg je 1 minuut extra speeltijd en wordt alles wat moeilijker. Jammer alleen dat het programma niet geheel vlekkeloos is. Er staan een paar spelfouten in, bijvoorbeeld "Alblasserdam" staat er verkeerd in. Van een dergelijk programma verwacht je een goede "scrolling". Dit valt wat tegen, daar de scrolling niet vloeiend, maar hortend is. Na een tijdje spelen is dit erg storend. Speel je het stedenspel, waar tijd erg belangrijk is, dan wordt je soms benadeeld. Zodra je 15 plaatsen goed hebt gevonden, wordt er met veel tamtam aangekondigd, dat je



"extra tijd" hebt verdiend. Al die fanfare kost je echter ook tijd, want de klok blijft doorlopen. Regelmatig zul je ondervinden dat de computer er lang over doet om je de volgende opdracht te geven, ook dat kost je tijd. Het programma stelt je tot slot nog in staat om je te overhoren. Een juiste spelling is daarbij dodelijk. Vijf fouten en je ligt eruit.

Maar alles bij elkaar is deze aanpak toch zoveel beter, dan wat we meestal aan educatief spul zien, dat we het van harte kunnen aanraden, je bent er (eventueel met de hele familie) avonden zoet mee, juist omdat het speldeel een extra uitdaging biedt.

Colossus Chess 2.0

Denkspelen per computer, het is een eeuwige uitdaging, maar bij dit programma komt de 64 een heel eind. Dit is werkelijk een fantastisch schaakprogramma! Je kunt er alle kanten mee uit.

Van de zeer vele mogelijkheden noemen we slechts een paar. De beginner die niets of weinig weet van schaken kan in een bepaalde situatie vragen wat je met een bepaald stuk kunt doen, dus waar deze heen kan en mag. Je kunt met dit programma tegen de computer spelen, maar ook spelen met 2 spelers. Vraag je het programma om advies in een bepaalde stelling, dan doet hij dat. Wil je weten of een schaakmat binnen een bepaald aantal zetten mogelijk is en hoe dat te bereiken is, dan zoekt hij dat voor je uit. Is een partij afgelopen, dan kun je de hele partij door de computer laten overspelen. Je kunt ook "blind spelen". Het programma denkt tot 14 zetten vooruit. Als de computer aan zet is, dan onderzoekt hij gemiddeld 520 mogelijkheden per seconde. De computer verslaan betekent dus van zeer goede schaakhuize komen. Andere mogelijkheden van dit programma, en dat zijn er vele, staan in een duidelijke handleiding. ●

Er zijn softwarehuizen, die vrijwel constant in de top van de hitlijsten aanwezig zijn. Activision is er een van, we noemen slechts het spel "Ghostbusters" van David Crane als voorbeeld.

Activision

Klasse-software

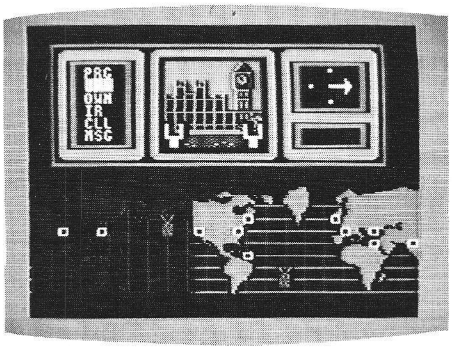
Toch heeft ook dit bedrijf het na een eerste succesronde met de videospelletjes (Atari 2600) erg moeilijk gehad en ook nu verliest men nog steeds geld. De overgang van spelmachine- naar microcomputer-software bleek namelijk niet gemakkelijk te zijn.

In London spraken we met de de marketing mensen van Activision, die druk bezig zijn om een eigen organisatie in Europa op te zetten. Ze lieten een deel van de nieuwe produkten voor het komende seizoen zien, die ze onder de aanduiding **Next Generation** gaan brengen. Sommige kenden we al uit de VS, maar er zitten nogal wat potentiële winners bij, dus geven we een kort overzicht en een uitgebreider commentaar op Baseball.

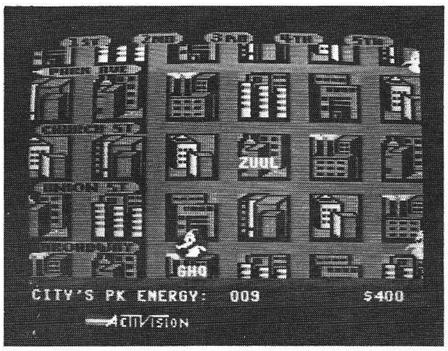
Mark Turnell ontwierp **Fast Tracks: Slot Car Construction Set**, een stukje lego-software om zelf raceauto's en racecircuits te ontwerpen en er dan ook mee en op te racen.

Met **Gamemaker** van Garry Kitchen krijgt men een programma om programma te maken. Middels eenvoudige menu's en instructies (in het Engels) kan men figuren, kleuren, sprites, animatie en acties kiezen en daarmee zijn eigen actiespel ontwerpen. De resultaten komen weliswaar niet uit boven ééndimensionale ontwerpen, maar men kan eindeloos variëren.

Met **Hacker** kan men zich op duistere paden begeven. Dat wil zeggen, in dit spel



gebeurt dat op legale wijze, maar het proces van zoeken naar computer-entry points, speuren naar codewoorden, en ook nog de wereld redden, is heel origineel verpakt. Zo krijgt men geen enkele documentatie, het spel begint met LOGON PLEASE en daar moet je het mee doen. Een adventure met onverwachte kanten en heel goede ondersteunende graphics.



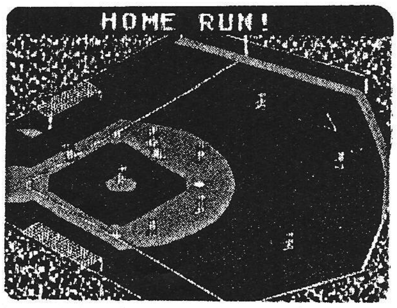
Ghostbuster was de eerste Activision computer-hit

Baseball Star League

In de Verenigde Staten is Baseball een van de meest populaire sporten. Dat is dan ook duidelijk aan dit spel te merken. Ik weet niet hoeveel uren de programmeurs, Dan Urgin, Scott Orr, en Bruce Mitchell aan dit spel hebben gewerkt, maar het moeten er vele geweest zijn. Het spel is een van de allerbeste voor de Commodore-64.

Het speelveld wordt driedimensionaal weergegeven. Tussen de innings door verschijnen er op het scorebord verschillende informaties. Dit kan behalve het toeschouwersaantal ook de tussenstand van een andere wedstrijd betreffen. Een typisch amerikaans trekje is dat er ook reclame op het bord wordt weergegeven. Het betreft hier reclame voor het spel On-Court Tennis hetgeen ook door Gamestar wordt uitgegeven. Behalve de fraaie graphics zijn de geluidseffecten ook bijzonder. Als u een bal een beetje verkeerd op de knuppel krijgt, hoort u ook een droge tik. Verder hoort u behalve de ingehuurde orgelspeler, die u bij de tussenstanden van muziek voorziet, ook nog het publiek, dat enthousiast juicht bij goede resultaten.

Het spel begint met het spelen van het Amerikaanse volkslied. Mocht u dit na de vele olympische uitvoeringen van dit werk niet zo bevallen, geen probleem, u kunt het volkslied door middel van de f 3 toets afbreken. Hierna krijgt u een keuzemenu op het scherm. U kunt dan instellen of u wilt oefenen of echt spelen. Voor een beginner is oefenen wel nodig, daar het driedimensionale effect het slaan behoorlijk bemoeilijkt. U kunt vervolgens instellen of u tegen een menselijke tegenstander of de computer wilt spelen. Daarna volgt nog de keuze van de werper en werpwijze. De werper is zo'n beetje de belangrijkste speler uit het team. U kunt kiezen uit 'Heat' Muldoon of uit 'Curves' Cassidy. Ze zijn beide verschillende werpers met hun eigen specialiteit. Later in de wedstrijd kunnen zij nog worden afgelost door 'Knuckles' Flanagan.



U bedient of de speler die aan slag is of het veldteam. Alle functies gaan met de joystick. Tot het spel behoren al de normale mogelijkheden zoals homeruns, ballen die wijd zijn geslagen, vangballen etc. U kunt een bal vanzelfsprekend ook foul slaan. Dit telt als een slag tenzij het uw derde slag betreft dan mag u meer ballen 'foul' slaan. Het doel van het spel is om de tegenspeler

SOFT WIR WAR

DOOR BERT TIER.



IDEE L. SALA.

Tekenen met de joystick is vaak knap moeilijk, omhoog en opzij gaat goed, maar diagonale bewegingen komen er niet lekker uit. Van Rinus van Elst uit Oss kregen we een oplossing voor dit probleem.

Hardware

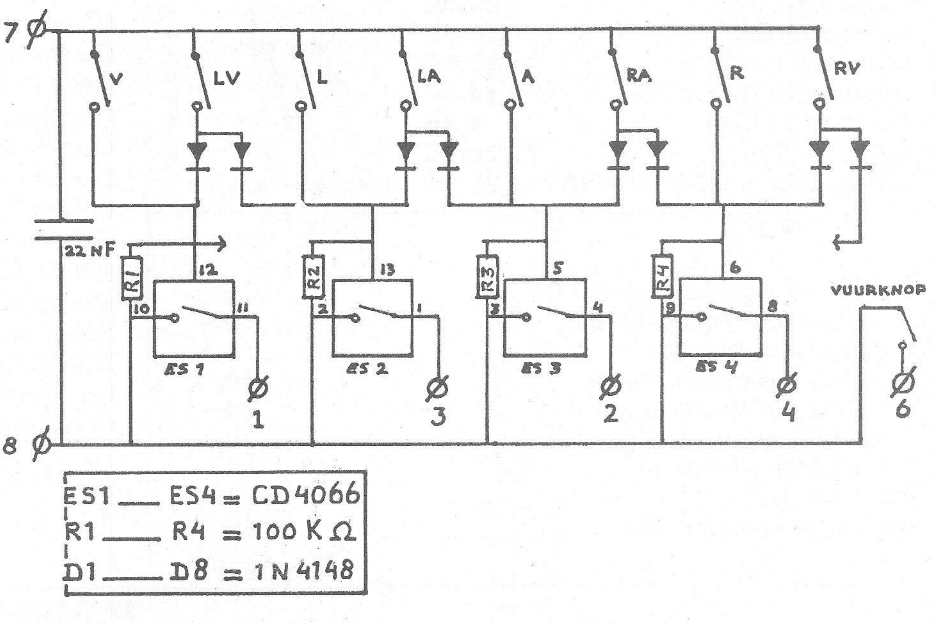
TEKENSTAAF

Een alternatief voor de joystick

Nogal wat tekenprogramma's gebruiken de signalen van de joystick om op het beeldscherm te tekenen. In eerste instantie lijkt dat een goede keus, maar na een tijdje merkt men, dat rechte lijnen wel, maar schuine niet echt lekker te zetten zijn.

Dat probleem ontstaat, omdat het met een gewone joystick vrijwel onmogelijk is om precies tegelijkertijd de twee schakelaars te sluiten, die samen de diagonale beweging moeten geven. Schuine lijnen worden dus in de praktijk trapgeveltjes en dat staat minder fraai. Nu is dat niet helemaal te verhelpen, de pixel- structuur van het beeld maakt er altijd een iets rafelige schuine rand van, maar het kan veel beter dan met de joystick.

Daartoe dient de onderstaand schakeling, waar vier electronische schakelaars de mechanische schakelaars in de joystick vervangen. Via een diodematrix worden ze geschakeld en dan zijn in plaats van vier richtingen nu drukknopsckaelaars beschikbaar voor 8 richtingen.



Die kun je aanduiden als: **Voor - Linksvoor - Links - Linksachter - Achter - Rechtsachter - Rechts - Rechtsvoor** of ook middels de streken van een kompasroos aanduiden, dus N - NO - O - ZO - Z - ZW - W - NW. De negende schakelaar in dit geheel dient als vuurknop. De diodes zorgen ervoor, dat bij horizontale of verticale bewegingen er maar één schakelaar gesloten wordt, terwijl bij diagonale beweging twee schakelaars tegelijk gesloten worden. De weerstanden R1 tm. R4 dienen om de stuuringangen van de schakelaars op het nul-potentiaal te brengen, wanneer er geen schakelaar is ingedrukt. De 22 nF condensator is er voor de ontkoppeling van de voedingsspanning, waardoor oscillaties voorkomen worden.

Het geheel, en daarbij kan de set schakelaars natuurlijk in een aangepaste opstelling worden gemonteerd, kan op een stukje montageplaat worden gezet. De beste opstelling blijkt een cirkelvormige montage van de schakelaars.

239+53=292

Basic Min

Een rubriek van Nico Baaijens me

Het gaat goed met deze rubriek en naar ik hoor, ook met de andere CI inzendingen. Rob vd Heuvel heeft zijn selectieploeg met drie man uitgebreid om alle inzendingen te kunnen verwerken, ze krijgen er werkelijk honderden. Ook bij mij komen steeds meer inzendingen binnen en zelfs is nu eindelijk ook gehoor gegeven aan mijn oproep om meer met geluid te doen en bijvoorbeeld de pingel van de STER-reclame na te bootsen.

Overigens staat deze rubriek nogal in het teken van de tekstbewerkingen. Uiterst leerzaam dus voor wie vlot in Basic overweg wil met de MID$, LEN() en andere tekstbewerkingscommando's.

Tien met griffel

Het wordt gewoonte de miniatuurtjes te openen met de begeerde onderscheiding: De Tien Met De Griffel voor de leukste, origineelste of meest bruikbare inzending. Dat moet een echte onderscheiding blijven, zodat ik nogal strenge eisen aan de dag leg, alvorens ik met tienen en griffels ga strooien.
Eén inzending, namelijk die van K. Schilstra uit Oudega, kwam er dicht in de buurt. Ik heb aan zijn aardige tekst-miniatuurtje nogal wat moet bijschaven, zodat ik iets afding op zijn Tien. Hij krijgt van mij een Tien-min, maar wél met de griffel voor deze inzending:

```
5 Q=1
10 POKE 53280,0: POKE 53281,0: PRINT"<WHT><SHIFT
   CLR/HOME>"
20 FORT=1TO30: READA,B: FORC=0TOQ: POKE 1024 +B
   +C *40,A: FORI=0TO50: NEXTI
30 POKE 1024+B+(C-1)*40,32: NEXTC: NEXTT
31 Q=Q+1: RESTORE: GOTO20
40 DATA14,0,9,1,3,2,15,3
41 DATA2,5,1,6,1,7,9,8,10,9,5,10,14,11,19,12
42 DATA2,14,1,15,19,16,9,17,3,18
43 DATA13,20,9,21,14,22,9,23,1,24
44 DATA 20,25,21,26,21,27,18,28,20,29,10, 30, 5, 31, 19, 32
```

C-16

Hans de Kruijf uit Culemborg beklaagt zich over het feit dat er zo weinig miniatuurtjes voor de C16 in deze rubriek staan. In elk geval is hij een van de eersten, die daar wat aan doet.
In de regels 40 en 50 zijn Commodore toets E en Comm.toets R opgegeven. Op de C16 stellen deze tekens respectievelijk voor: de omgekeerde T en de grafische T. Beide tekens zijn nodig om een gemetseld bakstenen muurtje te krijgen.

```
5 PRINT CHR$(147)
10 COLOR 0,1
15 COLOR 1,3,0
20 C=1: PRINT CHR$(5)
30 FOR K=1 TO 20: FOR D=1 TO 40
40 IF C=1 THEN PRINT "<Comm E>";: GOTO 60
50 PRINT "<Comm R>";
60 C=1-C
70 NEXT D
80 C=1-C
90 NEXT K
```

C-64 Input

Met de inzending van Henk van der Velden uit Schiedam wordt de INPUT-instructie met behulp van de GET-instructie nagebootst, compleet met prompting, cursor en het gebruik van de Return- toets. Voordeel: er kunnen nu ook komma's worden ingevoerd, zonder dat de Basic-interpreter daar tegen protesteert.

```
1 PRINT"INPUT? ";
2 POKE 204,0: POKE 207,0: GETWE$: IFWE$=""THEN2
3 POKE 204,1: IFWE$=CHR$(13)THEN5
4 IN$=IN$+WE$: PRINT WE$;: GOTO2
5 PRINT" ": PRINT IN$
```

Verrassing

Knotsgek is het miniatuurtje van Erik van Elven uit Leeuwarden. Ik verraad even niet wat het doet. Tik het zelf maar in en geef de C64 na RUN even de tijd om van binnen iets raars te doen. Het resultaat zal zeer verrassend zijn, al moet je wel even geduld hebben.

```
1 POKE 56334,PEEK(56334)AND254
2 POKE 1,PEEK(1)AND251
3 FORK=53248TO55295STEP8
4 FORT=K+7TOKSTEP-1: POKE 14336+A,PEEK(T): A=A+1:
NEXTT,K
5 POKE 1,PEEK(1)OR4
6 POKE 56334,PEEK(56334)OR1
7 POKE 53272,(PEEK(53272)AND240)OR14
```

Strooien met letters

Best aardig voor een Basic-programma met wat lettervuurwerk is de inzending van Hans-Jurgen Jacobs en Peter van der Putten uit Asten.

```
10 PRINT"<SHIFT CLR/HOME>": DIMA(100),C(100)
20 INPUT"WOORD OF ZIN: ";A$: PRINT"<SHIFT CLR/
   HOME>"
30 FORB=1TOLEN(A$): A(B)=ABS(ASC(MID$(A$,B,1))-64):
   NEXT: A=1
40 C(A)=INT(LEN(A$)*RND(1))+1: IFA=1THEN70
50 FORD=1TOA-1: IFC(A)<>C(D)THENNEXT: GOTO70
60 GOTO40
70 POKE 1504+C(A),A(C(A)): A=A+1: IFA=LEN(A$)+1 THEN
   END
80 GOTO40
```

iatuurtjes

orte tot zeer korte programma's.

Duits toetsenbord

Als een Duitser erover mocht klagen dat hij niet overweg kan met het Hollandse toetsenbord dan kan het hem in elk geval op de C64 naar de zin worden gemaakt met het miniatuurtje van Georges Nijs uit Testelt (België).

```
1 FORI=40960TO49151: POKE I,PEEK(I): NEXT: FOR
  I=14*4096 TO 65535: POKE I,PEEK(I): NEXT
2 POKE 1,53: T=60289: POKE T+10,ASC("Q"): POKE
  T+62,ASC( "A"): POKE T+12,ASC("W")
3 POKE T+9,ASC("Z"): POKE T+45,ASC("M"): POKE
  T+36,ASC(": ")
```

Aardig plaatje Leuke plaatjes kun je ook heel simpel op het scherm krijgen, zonder ingewikkeld gedoe met sprites. Laurent Buijk en Marc Gabriels uit Roosendaal geven daar een demootje van:

```
5 PRINT"<SHIFT CLR/HOME>": POKE 53281,1: POKE
  53280,1: PRINT"<CTRL ↑ (kwadraat)>": B=19: C=1
10 FOR E=1TO16: FORA=1TOC: PRINTTAB(B+A)"/";: NEXT:
   C=C +2: B=B-1
20 PRINT: NEXTE: B=B+3: C=C-6: IFC<14THEN10
30 FORD=1TO4: PRINTTAB(17);"<6 x SHIFT M>": NEXTD
40 GETA$: IFA$=""THEN40
```

Sorteersnelheden

Sorteren in Basic kan een langdurige geschiedenis zijn, hoewel de ene sorteerroutine sneller is dan de andere. Om te zien welke routine het er het snelst van afbrengt, kan gebruik worden gemaakt van het miniatuurtje (een subroutine) van Johan de Jong uit Leeuwarden.

```
1000 PRINT"BESTAND WORDT GESORTEERD"
1010 TI$="000000"
1020 REM GOSUB SORTEERROUTINE
1030 TD=VAL(TI$)
1040 UT=INT(TD/10000): REM UREN
1050 MT=INT((TD-UT*10000)/100): REM MINUTEN
1060 SE=TD-UT*10000-MT*100: REM SECONDEN
1070 PRINT"BESTAND IS GESORTEERD"
1080 PRINT"SORTEERTIJD WAS"UT"UREN"MT"MIN.EN"SE
     "SEC."
1090 END
```

Typing tutor

Voor computers zijn diverse programma's in de handel, die de gebruiker typen leren. Het geld voor zo'n typing tutor kan in de portemonnee blijven na het intikken van het miniatuurtje van Peter van den Brande uit Laakdal in België. Gebruikers kunnen de DATA-statements veranderen en woorden of letter/cijfer- combinaties invoeren, waarin zij vaak typo's maken.

```
15 PRINTCHR$(147)
20 FORI=1TO8: READA$
25 PRINT"TIK NA...: ";A$
30 TI$="000000"
35 INPUTB$
40 IFB$<>A$THENPRINT"FOUT...": GOTO35
45 PRINTTI$
50 NEXTI
100 DATA EEN,MEER,BOEKEN,ZOMER, COMPUTER, COM
    MODORE, TECHNIEK, QWERTY-TOETSENBORD
```

Ster-pingel

Mijn uitdaging om met geluidseffecten te werken en bijvoorbeeld de djingle van de STER-reclame na te bootsen, is opgepakt door Rinus van Elst uit Oss. In hoeverre hij erin geslaagd is, kan iedereen horen, die zijn miniatuurtje intypt.

```
10 S=54272: FORL=0TO24: POKE S+L,0: NEXT
20 PRINTCHR$(147): POKE S+24,15: POKE S+5,16: RE
ADH,L,D
30 IFH=-1THEN50
40 POKE S+1,H: POKE S,L: POKE S+4,17: FORT=1TOD:
   NEXT: POKE S+4,16: GOTO20
50 PRINT"HERHALING: TOETS": POKE 198,0: WAIT198,1:
   RESTORE: GOTO20
60 DATA126,151,30,119,124,30,126,151,30, 100, 121, 70, 75,
   69, 20, 75,69,20,75
70 DATA69,20
80 DATA84,125,20,100,121,20,112,199,20,126,151,20,-1,-1,-1
```

Prestatielijst

In veel dure games in machinetaal komt een prestatielijst voor, ook wel de 'Hall of Fame' genaamd, waarin de namen van top-scoorders worden vastgelegd. Ook voor Basic-programma's kan dat nuttig zijn, vandaar dat Wilfried Laureyssens uit Borgerhout in België dat in een aardig miniatuurtje verwerkte. Met de cursor kan langs letters en tekens worden gegaan en gekozen tekens worden met Return van het scherm geplukt. Pijl omhoog betekent stoppen.

```
10 PRINTCHR$(147)"(2 x spatie) ↑ ABCDEFGHIJKLMNO
   PQRSTUVWXYZ.!?&*/← " : POKE 646,6: P=1194
20 PRINTCHR$(19);: INPUTA$: A=PEEK(206)
30 IFA=31THENPOKE P,32: P=P-1: GOTO20
40 IFA=30THENPOKE 646,14: END
50 P=P+1: POKE P,A: POKE P+54272,2: GOTO20
```

Tot de volgende keer.
N.B.

UTILITIES 1

Speedsave 64

Hiermee kan de directory van een disk op het scherm komen, zonder het programmageheugen te verstoren. Ook versneld laden en saven naar disk via LOAD en SAVE en VERIFY, maar daarbij wordt de cassette-aansturing wel uitgeschakeld.

```
  1 rem speedsave
  2 rem door harrie de ceukelaire
  3 rem buizingen
  4 rem
 10 sys(peek(45)+256*peek(46)-202
    7)
 20 print"[CLR-HOME][RVS-aan][2xspati
    e]speedsave - 1541[22xspatie]
    [RVS-uit]";
 30 print"[neer]* enable[6xspatie]
    sys690
 40 print"[neer]* disable[5xspatie]
    run/stop restore
 50 print"[neer]* directory[3xspatie]
    load"chr$(34)"$ (naar scherm)
 60 print"[neer]* save[7xspatie]
    save"chr$(34)"naam
 70 print"[neer]* load[8xspatie]l
    oad"chr$(34)"naam
 80 print"[neer] (laden op plaats
     vanwaar gesaved)
 85 print"[neer]* load[8xspatie]l
    oad"chr$(34)"naam"chr$(34)",8
 90 print"[neer] (laden op begin
    basic ram)
 92 print"[2xneer][2xspatie]print
    er afzetten tijdens
 94 print"[neer][2xspatie]load/sa
    ve/verify !!
 96 new

        *** EINDE LISTING ***
```

Rom-Bug weg 64

Met sommige oudere 64's, waarin een oudere ROM zit, werken bepaalde programma om een of andere reden niet, zoals bv. Memory, Adresbak en Blauwe Bonen uit dit nummer. Om nu dit probleem te elimineren, is er deze utility, die dan (zolang u de machine niet uitzet) de bug opheft.

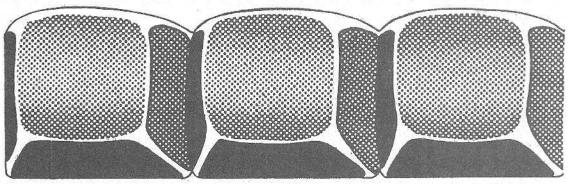
```
  1 rem bug weg / cbm-64
  2 rem door ronald willemsen
  3 rem uit bussum
  4 rem
 10 ifpeek(58588)=2thenprint"u heeft n
    ieuwe rom":end
 20 forx=828to931
 30 reada:pokex,a:next
 40 sys(828)
 50 end
100 data 165,113,141,170,3,165,114
110 data 141,171,3
120 data 160,0,169,0,133,113,169,160
130 data 133,114,177,113,145,113,200
140 data 208,249,230,114,165,114,201
150 data 192,208,241,160,0,169,224
160 data 133,114,177,113,145,113,200
170 data 208,249,230,114,165,114,201
180 data 0,208,241,169,134,141,219
190 data 228,169,2,141,220,228,165,1
200 data 41,5,133,1,160,0,169,4,133
210 data 114,169,32,145,113,200,208
220 data 249,230,114,16[illegible],114,201,8
230 data 208,241,173,170,3,133,113
240 data 173,171,3,133,114,96

        *** EINDE LISTING ***
```

```
bug weg                   regel 120     177
                          regel 130     173
                          regel 140     180
regel 1      232          regel 150     87
regel 2      51           regel 160     173
regel 3      96           regel 170     180
regel 4      143          regel 180     86
regel 10      78          regel 190     184
regel 20     110          regel 200     119
regel 30      26          regel 210     129
regel 40     145          regel 220     82
regel 50     128          regel 230     78
regel 100     79          regel 240     98
regel 110     61          ready.
```

Funktietoetsen

Zet onder de Funktietoetsen diverse funkties zoals LIST, PRINT, DATA en RUN. De funktiestoetsen kunnen alleen of in combinatie met de CTRL, Commodore LOGO, of Shift toetsen worden gebruikt. Op een balk op het scherm worden de mogelijke funkties weergegeven, maar dat betekent wel dat soms bij andere programma's die balk irritant in beeld is. NB: POKE 650,255 maakt alle toetsen repeterend en SYS 64738 Reset de computer.
Door Rene' van den Boom uit Langenboom.



```
  1 rem funktietoetsen / cbm-64
  2 rem door rene van den boom
  3 rem uit langenboom / 08863-1455
  4 rem
 10 print"[CLR-HOME][4xneer]";chr$(14);
    "[neer] Eventjes Geduld"
 20 p=49152:gosub60:p=49180:gosub60:p=4
    9200:pokep-1,32:forx=1to4:reada$
 30 fory=1to34:pokep,asc(mid$(a$,y,1)):
    p=p+1:nexty,x:p=49340:gosub60:sys49
    152
 40 print"[CLR-HOME][4xneer] Uitschakel
    en met Run/Stop + Restore [neer][4xs
PAtie]Inschakelen[2xspatie]";
 50 print"met SYS 49152":print"[2xneer]
    new[3xop]":end
 60 reada:ifa<>-1thenpokep,a:p=p+1:goto
    60
 70 return
```

UTILITIES 2

```
 80 :
 90 rem"[2xspatie]*** INITIALISATIE ***
100 data120,169,192,141,21,3,169,188
110 data141,20,3,169,192,133,252
120 data133,254,88,96,-1
130 :
140 rem"[3xspatie]***[2xspatie]VECTOREN
    [2xspatie]***
150 data48,57,67,76,82,89,102,110,116
160 data125,135,144,150,158,169,178,-1
170 :
180 rem"[3xspatie]***[2xspatie]WOORDEN[2
 xspatie]***
190 data"list←[4xspatie]print.[4xspatie]
    data.[4xspatie]run←[2xspatie]"
200 data"list.[2xspatie]poke650,255← re
    m.[4xspatie]run.[2xspatie]"
210 data"for.[5xspatie]next.[5xspatie]p
    oke.[4xspatie]peek(."
220 data"new.[4xspatie]sys64738.[2xspat
ie] save.[4xspatie]load. "
230 :
240 rem"[4xspatie]*** PROGRAMMA ***
250 data120,8,72,138,72,152,72,173,141,
    2,240,12,201,1,240,19,201,2,240,26,
    201
260 data4,240,33,169,46,32,41,193,189,2
    8,192,24,144,30,169,80,32,41,193,18
    9,32
270 data192,24,144,19,169,114,32,41,193
    ,189,36,192,24,144,8,169,148,32,41,
    193
280 data189,40,192,133,253,160,0,177,25
    3,201,46,240,23,201,95,208,12,169,1
290 data141,119,2,169,1,133,198,24,144,
    7,32,210,255,200,24,144,227,104,168
    ,104
300 data170,104,40,76,49,234,234,234,23
    4,133,251,160,33,177,251,201,46,240
    ,4
310 data201,95,208,2,169,32,41,63,9,128
    ,153,3,4,169,3,153,3,216,136,208,22
    9
320 data165,197,201,1,240,23,201,3,144,
    14,201,7,176,10,41,3,170,197,255
330 data240,3,133,255,96,104,104,24,144
    ,184,169,255,133,255,24,144,244,-1

              *** EINDE LISTING ***
```

syntax checksum: funktietoetsen

regel 1	95	regel 160	83
regel 2	214	regel 170	58
regel 3	151	regel 180	75
regel 4	143	regel 190	185
regel 10	21	regel 200	87
regel 20	96	regel 210	33
regel 30	216	regel 220	195
regel 40	196	regel 230	58
regel 50	215	regel 240	211
regel 60	153	regel 250	163
regel 70	142	regel 260	20
regel 80	58	regel 270	235
regel 90	246	regel 280	104
regel 100	243	regel 290	243
regel 110	34	regel 300	148
regel 120	162	regel 310	112
regel 130	58	regel 320	145
regel 140	19	regel 330	6
regel 150	45	ready.	

1541 Device Nr.

Wim Wubs uit Stadskanaal maakt het gebruik van andere diskdrives makkelijk en zet het devicenr. van de diskdrive softwarematig om.

```
1 print"[CLR-HOME][RVS-aan]device-num
  mer veranderen voor 1541"
2 print"[2xneer]schakel alle overige
  drives uit"
3 input"[3xneer]oud device-nummer";d1
4 input"[neer]nieuw device-nummer";d2
5 open1,d1,15
6 print#1,"m-w"chr$(119)chr$(0)chr$(2
  )chr$(d2+32)chr$(d2+64):close1
7 print"[2xneer]schakel de drives wee
  r in"

        *** EINDE LISTING ***
```

Converters/Decoders

Dit zet decimale getallen om naar binair en Hexadecimaal; Hexadecimaal naar Binair en Decimaal; en Binair naar Hexadecimaal en Decimaal, dus in wezen gelijk aan Decoder/hex, maar de aanpak verschilt. Door R. Beumer, Weesp.

```
  1 rem converters en decoders / cbm 64
  2 rem door dhr. r.beumer
  3 rem uit weesp / 02940-13670
  4 rem
 10 print"[CLR-HOME]";chr$(142)"[GEEL][RVS-
    aan][9xspatie]converters en decoder
    s[9xspatie][RVS-uit]":poke53280,7
 20 poke53281,0:p$="[3xneer][12xrechts]"
    :forq=0to62:reads:poke832+q,s:next:
    poke2040,13
 30 print"[HOME]"tab(252)"[l.groen]deci
    maal[5xspatie]= binair":printp$"bin
    air[7xspatie]= decimaal"
 40 printp$"decimaal[5xspatie]= hexadec
    imaal":printp$"hexadecimaal = decim
    aal"
 50 printp$"[GROEN]programma beeindigen
    [HOME][3xneer][rechts]";:forj=0to37
    :print"[GEEL][rechts][links]_";:next
    :c=7
 60 v=53248:w=68:g=54272:poke650,0:poke
    646,c:s=215:gosub260:print"[HOME][3xnee
    r]":fori=1to5
 70 print" ___":print" |M[2xspatie]M":print
    " MO¯¯| ":print"[2xspatie]¯¯¯":next:
    print"[HOME][5xneer][2xrechts]f1"
 80 t$="[3xneer][2xrechts]":printt$"f3"
    :printt$"f5":printt$"f7":printt$"f2"
    :c=c+1:ifc>15thenc=7
 90 forj=1to125:geta$:ifa$=""thennext:goto
    60
100 ifasc(a$)<133orasc(a$)>137thengosub
    330:goto60
110 ifa$="[ F2 ]"thengosub350:poke53269
    ,0:print"[CLR-HOME]"tab(216)"[GEEL]
    [RVS-aan] einde [RVS-uit]":gosub330
    :end
120 e=57:f=49:onasc(a$)-132gosub130,170
    ,200,230:s=215:gosub260:gosub350:clr
    :c=7:goto60
130 s=93:w$=" decimaal[4xspatie]":gosub
    260:r=val(b$):ifr<0orr>65535thengosub
    330:goto130
```

UTILITIES 3

```
140 c$(0)="0":c$(1)="1":r2=r:r$=""
150 t2%=r2/2:s=r2-t2%*2:r$=c$(s)+r$:r2=
    t2%:ift2%<>0then150
160 l=len(r$):t1=39-l:print"[HOME][2xneer]"
    tab(t1)r$:return
170 e=f:s=125:w$=" binair[7xspatie]":gosub
    260:n=len(b$):ifn<1orn>16thengosub3
    30:goto170
180 t1=0:c=0:forx=nto1step-1:c=c+1:t1=t
    1+val(mid$(b$,c,1))*2↑(x-1):nextx
190 l$=str$(t1):l=len(l$):t=39-l:print"
    [HOME][2xneer]"tab(t)t1:return
200 s=156:w$=" decimaal[4xspatie]":gosub
    260:a=val(b$):ifa<0ora>65535thengosub
    330:goto200
210 x=a/4096:print"[HOME]";tab(115);:for
    j=1to4:y=int(x):printchr$(y+55+7*(y
    <10));
220 x=(x-y)*16:nextj:print:x=a:return
230 e=70:s=188:w$=" hexadecimaal":gosub
    260:n=len(b$):ifn<1orn>4thengosub33
    0:goto230
240 forj=1ton:y=asc(mid$(b$,j)):z=z*16+
    y-48+7*(y>57):nextj:ifz>65535thenreturn
250 l$=str$(z):l=len(l$):t=39-l:print"[HOME
    ][2xneer]"tab(t)z:return
260 pokev+21,1:pokev+39,10:pokev+23,1:poke
    v+29,1:pokev,w:pokev+1,s
270 ifs=215thenreturn
280 print"[HOME][2xneer][39xspatie][CYAAN]"
    :b$="":print"[HOME][2xneer]"w$;
290 geta$:ifa$=""then290
300 ifa$=chr$(13)thenreturn
310 ifa$="="ora$="@"ora$=":"ora$=";"orasc
    (a$)<48orasc(a$)>ethengosub330:goto
    290
320 b$=b$+a$:print"[HOME][2xneer][14xrechts
    ]";b$:goto290
330 pokeg+24,15:pokeg+4,0:pokeg+4,33:poke
    g+6,240:pokeg+1,0:pokeg,0:pokeg+4,0
340 pokeg+4,17:fori=99to0step-1:pokeg+1
    ,i:next:pokeg+1,0:return
350 pokeg+1,0:pokeg,0:pokeg+4,0:pokeg+5
    ,0:pokeg+6,0:pokeg+24,15:pokeg+6,24
    0
360 pokeg+4,9:pokeg+4,17:fori=0to255step
    5:pokeg+1,i:nexti:pokeg+1,0
370 fori=255to0step-5:pokeg+1,int(rnd(1
    )*255):next:pokeg+1,0:pokeg+4,0:return
380 data248,0,0,127,128,0,15,224,0,3,24
    6,216,0,255,252,0,127,252,0,223,254
    ,0,239
390 data254,28,239,255,127,31,255,255,2
    55,255,227,255,255,0,127,255,0,31,2
    54,0
400 data15,254,0,7,254,0,3,254,0,1,254,
    0,1,254,0,1,254,0,1,254:rem r.beume
    r weesp,
```

*** EINDE LISTING ***

syntax checksum: converters

regel 1	225	regel 80	63
regel 2	15	regel 90	156
regel 3	97	regel 100	188
regel 4	143	regel 110	130
regel 10	253	regel 120	232
regel 20	2	regel 130	59
regel 30	1	regel 140	192
regel 40	98	regel 150	160
regel 50	237	regel 160	55
regel 60	241	regel 170	185
regel 70	66	regel 180	249

regel 190	10	regel 310	142
regel 200	54	regel 320	110
regel 210	15	regel 330	41
regel 220	240	regel 340	12
regel 230	78	regel 350	246
regel 240	39	regel 360	107
regel 250	126	regel 370	68
regel 260	120	regel 380	60
regel 270	93	regel 390	12
regel 280	73	regel 400	69
regel 290	104		
regel 300	83	ready.	

Regelteller

Heel fraai, deze toevoeging aan de reeks Basiccommando's, waarmee het aantal Bytes en regels van een programma snel kan worden bepaald en de listing worden gestopt. Machinetaal, dus commentaar over de structuur hebben we niet. Heel handig, ook voor de redactie van Commodore-Info om de lengte van een programma snel te bekijken.

```
  1 rem regelteller / c-64
  2 rem door arnoud van de wal
  3 rem uit apeldoorn / 055-66225
    8
  4 rem
 10 poke53281,0:print"[CLR-HOME][WIT]
    [4xspatie]_____________________
    _______"
 20 print"[4xspatie][RVS-aan]- re
    gelteller + liststop -
 30 print"[GEEL][neer] dit hulp-p
    rogramma telt het
 40 print" aantal regels in een b
    asic-
 50 print" programma en print dit
     met
 60 print" de exakte lengte uit.
 70 print" op te roepen door:
 80 print"[WIT][neer] regelprint
    (+return)
 90 print"[neer][GEEL] verder sto
    pt de listing door
100 print" het indrukken van shif
    t en
110 print" vertraagt bij indrukke
    n control.[op][WIT]
200 forz=52992toz+191:reada:pokez
    ,a:s=s+a:next
210 ifs<>25525thenprint"[2xneer]d
    ata-error":stop
220 a$="!eju!qsphsbnnb!cfwbu!!
    qsphsbnnb.!sfhfmt!fo!jt!!c
    zuft!pg!!nby"
230 a$=a$+"!!ljmpczuf)t*!mboh/
    !cz!b/h/!wbo!efs!xbm"
240 u=52735:fort=1to112:b=asc(mid$
    (a$,t,1))
250 pokeu+t,b-1:next:sysz-192
260 regelprint
500 data169,207,141,1,3,141,7,3,1
    69,19,141,0,3,169,155,141,6,3
    ,96,138,16
501 data3,76,139,227,32,121,0,201
    ,153,208,246,164,44,166,43,20
    8,1,136,202
```

UTILITIES 4

```
502 data134,251,132,252,160,0,132
    ,253,132,254,177,251,208,25,1
    60,2,177,251
503 data240,27,24,165,251,105,4,1
    33,251,144,2,230,252,230,253,
    208,2,230,254
504 data160,0,230,251,208,223,230
    ,252,208,219,162,0,32,143,207
    ,165,254,166
505 data253,32,205,189,162,24,32,
    143,207,165,45,56,229,43,170,
    165,46,229
506 data44,72,32,205,189,162,52,3
    2,143,207,104,74,74,170,232,1
    69,0,32,205
507 data189,162,70,32,143,207,162
    ,128,108,0,3,189,0,206,240,6,
    32,210,255
508 data232,208,245,96,72,169,1,4
    4,141,2,208,249,173,141,2,41,
    6,240,18,138
509 data72,152,72,162,48,160,160,
    136,208,253,202,208,248,104,1
    68,104,170
510 data104,76,26,167
```

regelteller

regel 1	207	regel 230	55
regel 2	222	regel 240	217
regel 3	88	regel 250	105
regel 4	143	regel 260	8
regel 10	77	regel 500	213
regel 20	11	regel 501	254
regel 30	113	regel 502	46
regel 40	44	regel 503	81
regel 50	72	regel 504	43
regel 60	229	regel 505	235
regel 70	42	regel 506	6
regel 80	41	regel 507	205
regel 90	163	regel 508	62
regel 100	61	regel 509	219
regel 110	208	regel 510	15
regel 200	174		
regel 210	134		
regel 220	76	ready.	

Splitscreen 64

Machinetaal-utility om je scherm in twee delen op te breken, waarbij de onderste helft in normale tekstmode blijft en de bovenste helft in Hi-res mode komt. Erg leuk om bijvoorbeeld tekst-beeld adventures of educatieve programma's mee te maken. Aanstuurcommando's in het programma.

```
  1 rem split-screen
  2 rem door h. groenewoud
  3 rem papendrecht / 078 154503
  4 rem
150 fora=49152to49295:readd:pokea
    ,d
151 t=t+d:next
152 ift<>18136thenprint"[CLR-HOME]
     fout in data":stop
200 print"[CLR-HOME][neer] dit pr
    ogramma zet de bovenste helft
    [4xspatie][neer] van het sche
    rm om in ";
210 print"high-resolution."
220 print"[2xneer] gebruik [RVS-aan]
     sys49152 [RVS-uit] om de rou
    tine te[3xspatie][neer][2xspatie]
    starten."
230 print"[2xneer] gebruik [RVS-aan]
     sys49253,kleur [RVS-uit] om
    het hires- [neer] veld schoon
     te vegen";
240 print" en de veldkleur[3xspatie]
    [neer] te zetten."
250 print"[2xneer] hires-schermwa
    arden :"
260 print"[neer] x : 0 - 319[10xspati
    e]normaal : 0 - 319"
270 print"[neer] y : 0 -[2xspatie]
    94[10xspatie]normaal : 0 - 19
    9"
280 fort=1to10000:next:print"[2xop]"
299 end
300 rem machinecode data
310 data120,169,127,141,013,220,1
    69,001
311 data141,026,208,169,000,141,0
    18,208
312 data173,017,208,041,127,141,0
    17,208
313 data169,036,141,020,003,169,1
    92,141
314 data021,003,088,096,173,025,2
    08,141
315 data025,208,041,001,240,028,1
    73,018
316 data208,201,145,240,024,173,0
    24,208
317 data009,008,141,024,208,173,0
    17,208
318 data009,032,141,017,208,169,1
    45,141
319 data018,208,076,049,234,173,0
    17,208
320 data041,223,141,017,208,173,0
    24,208
321 data041,247,141,024,208,169,0
    00,141
322 data018,208,076,188,254,032,2
    41,183
323 data142,133,192,162,015,169,0
    00,133
324 data251,169,032,133,252,160,0
    00,152
325 data145,251,200,208,251,230,2
    52,202
326 data208,243,160,000,169,008,1
    53,000
327 data004,153,224,004,200,208,2
    47,096
```

split-screen

regel 1	8	regel 260	184	regel 318	125
regel 2	56	regel 270	143	regel 319	138
regel 3	189	regel 280	66	regel 320	118
regel 4	143	regel 299	128	regel 321	117
regel 150	177	regel 300	185	regel 322	139
regel 151	4	regel 310	115	regel 323	119
regel 152	22	regel 311	120	regel 324	115
regel 200	100	regel 312	123	regel 325	111
regel 210	108	regel 313	125	regel 326	114
regel 220	244	regel 314	126	regel 327	120
regel 230	213	regel 315	114		
regel 240	89	regel 316	117		
regel 250	165	regel 317	123	ready.	

Print-out

Met o.a. Hit the ball - Sjoelen - Faculteit - Catch the ball - Gokspel C-16

Er is wat afgeprogrammeerd de laatste maanden, we zijn werkelijk bedolven onder de bandjes met programma's. Dank daarvoor, Commodore-Info kan dank zij uw inzendingen nog beter worden. Gezien de enorme massa inzendingen voor onze prijsvraag zullen een aantal inzenders teleurgesteld zijn, dat ze hun inzending niet of nog niet in Commodore-Info zien. Maar u kunt zich de taak voor Rob v.d. Heuvel en zijn team voorstellen, ze moeten door 200 of meer programma's heen en dat gaat dus ten eerste niet zo snel, maar de concurrentie is ook groter. Zo kunnen ze nu de beste kiezen uit veel meer materiaal. Iedere inzender heeft ondertussen bericht gehad en u hoort de volgende keer wie de Novex monitor gewonnen heeft en we stellen ook nog wat andere prijzen ter beschikking. Een bijeffect van deze prijsvraag is, dat we nu beschikken over veel meer korte programma's, die dus wat sneller in te tikken zijn, maar soms ook wat geleden hebben aan duideijkheid en structuur. In het algemeen moet men er bij inzendingen aan denken, dat veel vreemde (grafische) of reverse tekens tot problemen kunnen leiden, omdat onze printer er geen weg mee weet of ze niet 100% juist afdrukt, zoals met de reverse b, die tussen de haakjes in q1$="" in het programma Videotrack in regel 20050 had gemoeten. Dat soort dingen merken ook wij te laat, het programma draaide bij ons goed, maar de print-out was het dus niet helemaal.

INHOUD VAN DIT LISTINGDEEL

Hit the Ball	39	Tafels	53
Sjoelen 64	39	Ketting 64	54
Commodore Show	40	Machtsverheffen	55
Boter, kaas en eieren	42	Blauwe Bonen	56
Catch the Ball 64	51	Geldweg	56
Faculteit	52	Vic liststop	59
Beeldeffect	52	Vic ondersteboven	59

Checksum-listtester 64

Nog maar eens het Syntax-checksum programma om uw intikwerk te controleren. Intikken, saven, en dan via RUN onderin in het geheugen plaatsen, dan te testen programma laden (niet runnen) en via SYS 49152 testen.

Dit is de versie, waarbij met Run/stop de listing op het scherm kan worden gestopt en met F-1 of F-7 weer kan worden doorgegaan. Op papier printen gaat met : OPEN 4,4: CMD 4: SYS 49152 en dan na het printen weer PRINT #4: CLOSE 4 om het af te sluiten. Niet alle fouten haalt u hiermee uit uw programma's, want omwisselingen zoals uv ipv. vu komen er niet uit, maar vooral dataregels kunt u er heel goed mee controleren.

```
 1 rem ********************************************
   *******************
 2 rem                basic loader   "SYNTAX.CHECKS
   UM"
 3 rem  runnen en new.  blijft in ram op $c000.
 4 rem          checksum testen met commando  :
   sys 49152'
 5 rem
 6 rem v.840928.64   jan bodzinga
 7 rem ********************************************
   *******************
10 i=49152  :rem beginadres
20 reada:ifa<0then50  :rem data ingelezen
30 pokei,a:i=i+1:goto20
50 poke49184,148:poke49185,192
55 i=49300
60 read a: ifa<0then90
70 pokei,a:i=i+1:goto60
90 print"data is weggezet"
 95 print"checksum testen met sys49152"
100 data 165,  43, 166,  44, 133, 163, 134, 164,
    169, 147
110 data  32, 210, 255, 160,   0, 240,   3,  32,
     73, 192
120 data  32,  73, 192, 208,   1,  96,  32, 225,
    255, 208
130 data   3,  76, 116, 164,  32,  81, 192,  32,
     73, 192
140 data 240,  12, 201,  32, 240, 247,  24, 101,
    167, 133
150 data 167,  76,  37, 192, 166, 167, 169,   0,
    132, 168
160 data  32, 205, 189, 169,  13,  32, 210, 255,
    164, 168
170 data  76,  17, 192, 200, 208,   2, 230, 164,
    177, 163
180 data  96, 162,   0, 189, 123, 192, 240,   6,
     32, 210
190 data 255, 232, 208, 245,  32,  73, 192, 170,
     32,  73
200 data192, 132, 168,  32, 205, 189, 162,   3, 1
    69,  32
210 data  32, 210, 255, 202, 208, 250, 169,   0,
    133, 167
220 data 164, 168,  96,  82,  69,  71,  69,  76,
     32,   0
230 data  -1
240 data 165,197,201,3,240,7,201,4,240,6
250 data 76,148,192,76,34,192,169,147,32,210
260 data 255,76,161,192
270 data -1

             *** EINDE LISTING ***

Checksum: Checksum.nw
```

regel	1	249	regel	110	158
regel	2	84	regel	120	232
regel	3	6	regel	130	183
regel	4	117	regel	140	96
regel	5	143	regel	150	96
regel	6	17	regel	160	127
regel	7	61	regel	170	71
regel	10	157	regel	180	223
regel	20	65	regel	190	73
regel	30	203	regel	200	79
regel	50	14	regel	210	109
regel	55	251	regel	220	106
regel	60	193	regel	230	225
regel	70	207	regel	240	114
regel	90	245	regel	250	89
regel	95	237	regel	260	68
regel	100	183	regel	270	225

Hit the Ball

Spelletje met schieten op een bewegende bal.

```
  1 rem hit the ball / c-64
  2 rem door arnoud van de wal
  3 rem uit apeldoorn
  4 rem
 10 sys65409:fort=832to1023:poket
    ,0:next:poke855,60
 20 fort=862to880step3:poket+0,25
    5:reada:poket+34,a:poket+98,2
    8:poket+100,112
 30 poket-30,24:poket-28,24:poket
    +2,255:next:fort=883to895:poke
    t,255:next
 40 poke859,126:poke853,60:poke86
    1,126:fort=49152tot+94:reada:
    poket,a:next
 50 b$="[8xrechts]":a$=b$+"[22xspatie
    ]":c$="[HOME][4xneer]":print"
    [CLR-HOME][GROEN][RVS-aan]"chr$
    (8);
 60 printchr$(142);c$a$:fort=0to1
    3:printb$"[RVS-aan] "tab(29)"
     ":next:print"[RVS-aan]"a$
 70 c=53248:poke2040,13:poke2041,
    14:poke2042,15:pokec+39,7:poke
    c+40,1:pokec,150
 80 pokec+1,180:poke254,0:pokec+4
    1,1:pokec+32,6:pokec+33,0
 90 sys49152:poke767,0:pokec+21,7
100 v=54272:pokev+24,15:fort=0to2
    3:pokev+t,0:next:pokev+24,15:
    pokev+1,5
110 pokev+15,10:pokev+6,248::poke
    v+7,150:pokev+8,1:pokev+12,41
    :
120 printc$tab(20)"[WIT]AAAAA":print
    "[HOME] [RVS-aan][BLAUW] hit
    the ball (c) 1985 c-info
130 l=5:s=0:printb$"[neer][WIT] s
    core[4xspatie]ships left
140 x=int(rnd(0)*130)+100:y=90:q=
    peek(c+30)
150 fort=1to7:r=int(rnd(0)*3)-1:r
    =r*(s/40+1)
160 foru=1to15:x=x+r:y=y+1:ifx>24
    8orx<96thenr=-r:x=x+r:pokev+4
    ,19:pokev+4,18
170 pokec+2,x:pokec+3,y:k=peek(c+
    30)
180 ifk>0thenifk<>5thent=9:u=99:next
    :next:goto200
190 next:next:s=s-10:pokev+4,21:poke
    v+4,20:goto270
200 if(kand1)then230
210 pokec+5,1:pokev+8,1:pokev+11,
    128:pokev+11,129:fort=0to65:poke
    c+40,t:next
220 s=s+10:goto270
230 pokec+21,5:pokev+12,43:pokev+
    8,0:pokev+11,128:pokev+11,129
240 fort=0to6:pokec+17,24:pokec+3
    9,0:q=2↑20:pokec+17,31:pokec+
    39,1:q=2↑10
250 next:pokec+3,20:pokec+17,27:poke
    c+39,7:pokec+21,7:pokev+12,41
260 l=l-1:printc$tab(24-l)"[GROEN]
    [RVS-aan]V":ifl<.then290
270 printc$"[WIT]"tab(10);s"[links]
     ":on-(s<0)goto290:ifs>120then
    poke254,1
280 pokec+3,20:fort=0to500:next:goto
    140
290 poke788,92:printc$,"[5xneer][3xsp
    atie][WIT]game[2xspatie]over[l.bl
    auw]":pokec+21,0
300 print,"[2xneer][3xspatie]pres
    s fire":wait56320,16:wait5632
    0,16,16:run50
310 data126,231,219,195,219,219,1
    26
320 data120,169,13,141,20,3,169,1
    92,141,21,3,88,96,173,255,2,2
    4,109,0,208
330 data201,97,176,2,169,97,201,2
    31,144;2,169,231,141,0,208,16
    2,255,173,0
340 data220,168,41,4,240,11,162,1
    ,152,41,8,240,4,166,254,208,3
    ,142,255,2
350 data173,5,208,201,91,144,8,20
    6,5,208,206,5,208,208,18,162,
    5,152,41,16
360 data208,2,162,190,142,5,208,1
    73,0,208,141,4,208,76,52,234

          *** EINDE LISTING ***
```

hit the ball

regel 1	121	regel 170	74
regel 2	222	regel 180	51
regel 3	37	regel 190	95
regel 4	143	regel 200	67
regel 10	176	regel 210	207
regel 20	169	regel 220	191
regel 30	51	regel 230	105
regel 40	95	regel 240	9
regel 50	60	regel 250	146
regel 60	156	regel 260	7
regel 70	88	regel 270	152
regel 80	199	regel 280	131
regel 90	248	regel 290	70
regel 100	56	regel 300	88
regel 110	196	regel 310	198
regel 120	93	regel 320	3
regel 130	65	regel 330	8
regel 140	21	regel 340	190
regel 150	154	regel 350	2
regel 160	223	regel 360	178

Sjoelen 64

Reactiespel naar het oudhollandse sjoelbak principe, compleet met puntentelling. Lekker kort en de schermopmaak is dan ook niet denderend, maar dat kan men zelf verbeteren. De machinetaalroutines zijn deze keer niet met dataregels, maar van regel 40 tot 170 met pokes gedaan en dat is ook een manier.

```
  1 rem sjoelen / cbm-64
  2 rem door h.alphenaar
  3 rem uit alkmaar
  4 rem
 10 rem een druk op de spatiebalk
     doet de schijf naar rechts g
    aan . 3 beurten.
 20 s=0:p(a)=0:q(a)=0:r(a)=0:s(a)
    =0
 30 print"[CLR-HOME][5xneer][11xrecht
    s][RVS-aan]s j o e l e n[RVS-uit]"
    :forx=0to39:poke1304+x,67
 40 poke1784+x,67:nextx:forx=0to9
    :poke1454+x,67:poke1574+x,67:
    poke1694+x,67:next
```

```
 50 poke1334,114:poke1374,66:poke
    1454,107:poke1534,66:poke1574
    ,107:poke1614,66
 60 poke1694,107:poke1734,66:poke
    1814,113:poke1423,50:poke1543
    ,51:poke1663,52
 70 poke1783,49:poke1342,114:poke
    1382,66:poke1422,66:poke1462,
    91:poke1502,66
 80 poke1542,66:poke1582,91:poke1
    622,66:poke1662,66:poke1702,9
    1:poke1742,66
 90 poke1782,66:poke1822,113:be=4
    9:gosub370
100 n=1125:poken,19:poken+1, 3:poke
    n+2, 8:poken+3, 9:poken+4,10:
    poken+5,22
110 poken+6, 5:poken+7,14:gosub38
    0
120 for b=0to10:poke1344+40*b,87:
    poke1344+40*(b-1),32:geta$:if
    a$=chr$(32)goto160
130 ifa$<>chr$(32)thennextb:for b
    =10to0step-1:poke1344+40*b,87
140 poke1344+40*(b+1),32:get a$:if
    a$=chr$(32)goto160
150 if a$<>chr$(32)thennext b:goto
    120
160 s=s+1:gosub380:y=1345+40*b:poke
    y,87:forx=0to36:pokey+x,87:poke
    y+(x-1),32:
170 z=x+y+1:ifpeek(z)=66orpeek(z)
    =107orpeek(z)=87thengoto190
180 nextx
190 pokey+x,87:ifs=30goto210
200 goto120
210 lb=0:forr=0to10:forl=1to30:if
    peek(1343+40*r+l)=87thenlb=lb
    +1
220 poke1343+40*r+l,32:nextl:next
    r:s=30-lb:be=be+1:ifbe=52goto
    240
230 gosub370:gosub380:goto120
240 print chr$(19)"einde spel":for
    a=1to7:a(a)=peek(1422-a):b(a)
    =peek(1502-a)
250 c(a)=peek(1662-a):d(a)=peek(1
    782-a):if a(a)=87 then p(a)=2
260 if b(a)=87 then q(a)=3
270 if c(a)=87 then r(a)=4
280 if d(a)=87 then s(a)=1
290 nexta:f=0:fora=1to7:ifa(a)+b(
    a)+c(a)+d(a)=348 then f=f+10
300 f=p(a)+q(a)+r(a)+s(a)+f:nexta
    :print"[neer]het totaal is [RVS-a
    an]";f;"[RVS-uit]":iff>recthen
    rec=f
310 print"[6xneer]het record is :"
    ;chr$(13)"[RVS-aan]";rec;"[RVS-ui
    t]"
320 print"[neer]nog een spel ? j/
    n"
330 get a$:if a$=""then330
340 if a$="j"goto20
350 if a$="n"thenend
360 if a$<>"n"goto330
370 m=1958:pokem,2:pokem+1,5:poke
    m+2,21:pokem+3,18:pokem+4,20:
    pokem+6,be:return
380 sc=30-s:poken+11,32:printchr$
    (19) "[2xneer][30xrechts][RVS-aan
    ]";sc;"[RVS-uit]"
390 return

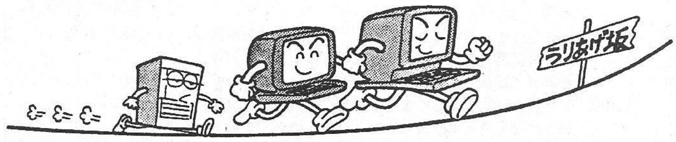
          *** EINDE LISTING ***
```

sjoelen

regel 1	55	regel 190	214
regel 2	197	regel 200	28
regel 3	122	regel 210	107
regel 4	143	regel 220	63
regel 10	183	regel 230	223
regel 20	51	regel 240	192
regel 30	241	regel 250	14
regel 40	95	regel 260	239
regel 50	144	regel 270	242
regel 60	87	regel 280	241
regel 70	46	regel 290	53
regel 80	6	regel 300	142
regel 90	161	regel 310	236
regel 100	175	regel 320	228
regel 110	25	regel 330	99
regel 120	238	regel 340	27
regel 130	217	regel 350	91
regel 140	213	regel 360	5
regel 150	146	regel 370	200
regel 160	85	regel 380	122
regel 170	1	regel 390	142
regel 180	218	ready.	

Commodore Show 64

Dit programma toont een aantal schitterende beeldeffecten, die ook nog worden begeleid door geluiden. Het woord Commodore schuift in beeld en dan gaat er van alles omheen gebeuren, de tekst beweegt, de kleuren veranderen etc. Leuk om te bekijken en technisch qua lengte binnen de voorwaarden van onze prijsvraag. Toch juichen we deze overcompacte manier van programmeren niet toe, het overtypen is moeilijk en de structuur is niet te herkennen.



```
 1 rem commodore show
 2 rem door jan van ee
 3 rem barneveld / 03420-15383
 4 rem
 5 print"[CLR-HOME][12xneer][8xspati
   e]2 minuten geduld a.u.b."
10 s=49152:z=s:forl=1to37:readd$
   :forb=0to23:p=asc(mid$(d$,3*b
   +3,1)):v=p+2*(p>57)
20 q=asc(mid$(d$,3*b+2,1)):v=41*
   v+q+2*(q>57):r=asc(mid$(d$,3*
   b+1,1)):v=41*v+r
30 v=v+2*(r>57)-82704:pokes+1,int
   (v/256):pokes,v-256*peek(s+1)
   :s=s+2:nextb,l:sysz
40 data1bq2ukjv4dboaa<hz478p?<pw
   av540h9pyihr<kebf?=kfbfksj6aj
   17izqk1aja3hdza@9k
50 dataxr3g=1zzw07xi2096x9g2er30
   7xrs096xmrqc6u==k1b6phvk5k1b6
   or37s9fnwguws80ptw
60 dataj7kibfc=kjbfosjcpk2=f5ej@
   yl8r5d7h@uflkwpkx80b0>fc=ks5b
   e<ff>ptqq9m8pkxhr3
70 datayjxzqx<u48qxmrqxq39fkdn2<
   ?x80bl<fa=kq5bd<favkh?ffo@540
   8769mv@pkkkgs<o540
80 datafnwxy4ptwirk>fgl>o4dezfkv
   ilwtkzegk?o2wlz5jcrqqpkgegl>o
   @jezfkvilwtkqfgk?o
```

```
 90 data<1hz5jwtkd<f@vk9=fiq@3q3=
    f1jeg?nqan2=f17fg6nqacj?>ibj0
    ?cuc>ivv0?cuxdpusv
100 datavk=>eo540cn0ijg6>ri=ulz4d
    bo18vamvvu4odogfvamvvu4rdoscu
    b3=a<0ts2bukwegs=r
110 data2juamvvu4udohvuamvvu4xdo2
    >d3yacup4qqwn=vbod6v7ib5>pehu
    7ib1>plsvuejoofrlx
120 datamrqeq@w30htk?=0acjcyldbix
    =p9g2wr3<tk<pieskfzki20oyklg5
    gz41bolgmesk8fooyk
130 datat155p@o46d<o8>o@@o=aof67j
    s3n40c4mn408txw460008>o000n@o
    6>2@s3n4057nn408tx
140 datasg10008>o000=ao6>23t3n405
    7nn408txv36d<o8>o7@on@o7imf00
    mq@n40bysi63786d<o
150 data000@@o000b4m3t3n40c4mn408
    tx<960008>o000=aokctf00mq@n40
    h>pi63=96d<o8>o7@o
160 datan@o960@ooytkdeggnqf45jrjs
    kgqa0xy43ow6jkt614v5uipztp4zv
    a92qp36ukmxipmq00j
170 dataomq80b6kkmrq2vkeeg5nq4deg
    f4ejg>nqi=3>ukkkgbnq>y51uk0@f
    btke<fctkg=faadn1v
180 data?53>pk5jg<e0i24>v0ceganqy
    w4<5pf31lz4<5p@je5qiijgs=rirk
    y>fbvk8afcvkd<f>vk
190 datal<f5gj7ylh9hujps1xisxj8xp
    ?o=>111x=0xonqdq@glk0ykpblgnq
    f45v9=geggnqm614v5
200 data@vp28xe<0xq3c<128pwqqn6pl
    lvnqky?fc9k09=lbfxqqfr5us5us5
    us5tc6xggqmq7h0ejg
210 dataumq661ejgymq5q1ejg2nqs84i
    jg<>r6y?u>12qqdwpivwt>d3i7?53
    >pk4kg<e0540r14>uk
220 dataqeg@nq0e?i10330=txr@2dley
    txfm9ukeh10@ydz?e130@@d<vdx20
    8@d000x208@d<vd130
230 data?ydz?eh10em9ukeytxr@2dle=
    tx530330000000000000000000000
    0004qx4e0ru44if>71
240 datay0n6001x0o>7600v31y0n4if?
    i0ru44qxa00000000000000000000
    000000000000000000
250 data00000000093o<ywio9t8uoh4?
    sorle?22koerle9f4koeymevzwkoe
    @re630000000000000
260 data0000000000000000000000000
    00000000v?p5f80ot2fxw6ny7d0c7
    <tl@<80c7<tljb80c7
270 data=tlkr70c78100000000000000
    00000000000000000>00xs1000>00
    xsluc2yyx?yu6on9f0
280 dataig4yrli60e32yrl2e0ig46onn
    ux?yuuc2q5000000000000000000
    000000000000000000
290 data000000000fhg@zxyrxrgv8<og
    81egb9fel20egbt9oq5023v8txz40
    <bg000000000000000
300 data0000000000000000000000000
    00000000b5taj9u508@d?2ca10jtw
    z?e260@20?2ca108@d
```

```
310 dataaj9u50b5t0000000000000000
    00000009pkgmjuwio=rth2?vkqeg
    >>r@jemt0ijg7>ru0x
320 datawz0ijg8>rk22bukweg6>rv764
    a2bukyegs=rl=>b>1nrj9aj?vkb?f
    u4ga4o8jgp=r8mk0>l
330 data>?a2qqdwpdx6srqdil0nka4o8
    jgp=r<mky=1@?amrq@ufbx6srqyok
    inkttfr>l4pp4xx6xx
340 datadx6@w5us5p@g3>rmrq4uf5avp
    16ejgtnq063cn0zz4e5ppgggmq7h0
    ejgumq661ejgymq5q1
350 dataejg2nqirk?af89knqqiq@pdr1
    ukmig>>r?xxn90ijgw=r2j3bukhhg
    s=r5qinrjmkg2>ripg
360 data3>rih2ts2buk6@flsjnbfcwpz
    jxfex9wtibf3ylb6ponvi5khkg9>r
    pkxsc3on=p6p<?x8gu
370 dataf<f<3h5kpyrlc@lz7p395=j<<
    p@f<0@vp7y4o4pholl z4<5py30gnw
    qtwf>xpdx>qg2>ripg
380 dataz=rxz0spg8>ripgp=rjpe2j3b
    uk6fg>>r<h93z2bukpegt=r6sk2jg
    9e0c<pi=3hb?16kzad
390 data3i7?53>pk4kg<e0540i=3>ukj
    v4dbolz4ejghnq3q33a6ckg?nqan2
    3a6tjg6nqyw4<5phbl
400 datab>rngileg<>r4a2on==mu540z
    zwxy496xgs4540w30=0x2qq97uawp
    w6w@20000000000000
410 "[5xDELete]
420 "[5xDELete][3xspatie]jan van
    ee
430 "[5xDELete][3xspatie]vliegers
    velderlaan 25
440 "[5xDELete][3xspatie]3771 xe[2xsp
    atie]barneveld
450 "[5xDELete][3xspatie]tel: 034
    20-15383
460 "[5xDELete]

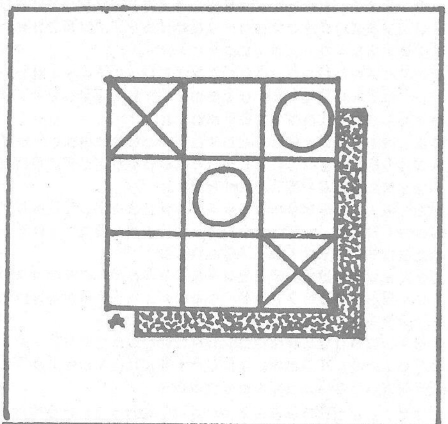

         *** EINDE LISTING ***

  commodore show


  regel 1     117         regel 230    149
  regel 2     11          regel 240    67
  regel 3     123         regel 250    123
  regel 4     143         regel 260    203
  regel 5     165         regel 270    222
  regel 10    225         regel 280    244
  regel 20    16          regel 290    241
  regel 30    24          regel 300    45
  regel 40    200         regel 310    72
  regel 50    225         regel 320    230
  regel 60    37          regel 330    234
  regel 70    240         regel 340    21
  regel 80    158         regel 350    144
  regel 90    255         regel 360    183
  regel 100   112         regel 370    31
  regel 110   248         regel 380    4
  regel 120   89          regel 390    54
  regel 130   107         regel 400    156
  regel 140   111         regel 410    134
  regel 150   254         regel 420    206
  regel 160   118         regel 430    44
  regel 170   22          regel 440    136
  regel 180   182         regel 450    207
  regel 190   89          regel 460    134
  regel 200   134
  regel 210   152
  regel 220   227         ready.
```

Boter, kaas en eieren

Als spel bij vrijwel iedereen bekend, op de computer toch wel aardig. De uitvoering is niet helemaal perfect, je kunt over een bezette plek toch heengaan, dat zou met een extra testroutine kunnen worden verholpen.



"BOTER-KAAS EN EIEREN"

```
  1 rem boter kaas en eieren / c-
    64
  2 rem kees wielaart
  3 rem uit dordrecht / 078 16282
    1
  4 rem
  5 rem
 10 p1=48:p2=48
 20 poke646,0:y=0
 30 poke53280,11:poke53281, 1
 40 printchr$(147)
 50 m=0
 55 n=0
 60 poke(1078+n+m),79:poke(1079+n
    +m),119:poke(1080+n+m),80
 65 poke(1118+n+m),116:poke(1120+
    n+m),106
 70 poke(1158+n+m),76:poke(1159+n
    +m),111:poke(1160+n+m),122
 75 n=n+3
 80 if n=3 or n=6 then goto 60
 85 m=m+120
 90 if m=120 or m=240 then goto 5
    5
100 poke1119,49:poke1122,50:poke1
    125,51
110 poke1239,52:poke1242,53:poke1
    245,54
120 poke1359,55:poke1362,56:poke1
    365,57
130 poke 1149,p1:poke1174,p2
140 fork=1065to1074:pokek,67:next
150 forl=1090to1099:pokel,67:next
155 poke214,0:poke211,1 :sys58732
    :print"speler (1)"
160 poke214,0:poke211,26:sys58732
    :print"speler (2)"
170 print"[10xneer]"
180 input"[12xrechts]speler (1)";
    a
190 z=86:s=1:goto 230
200 input"[op][12xrechts]speler (
    2)";a
210 z=87:s=2:goto 230
230 ifa=1 then poke 1119,z
240 ifa=2 then poke 1122,z
```

Eindelijk staat dan het grote moment voor de deur; de Commodore PC 128 computer zal binnenkort bij de dealers in de winkel staan. We hebben er maandenlang naar uitgekeken, en menig indianenverhaal over soft- en hard-bugs moeten verduren, maar nu is het dan zover. Als we Commodore moeten geloven, begint de levering nu toch echt begin oktober. Onze verwachtingen waren hoog gespannen en we zullen nu snel te weten komen, of de 128 het waar gaat maken. Op de Commodore Info beurs in Den Haag had Jan Bodzinga gelegenheid een paar van de (Belgische) 128-machines wat beter te bekijken en te beproeven. Hij geeft een voorbeschouwing over het wat trage antwoord van Commodore op de concurrentie.



C-128 : opvoerset hard nodig

Al snel nadat de C-64 in Nederland was geïntroduceerd werd duidelijk dat deze computer de trendsetter zou worden voor de stand van zaken in de huiscomputerwereld. Zelfs voor de originele prijs van ƒ 1300 zat er in deze computer een geheugen, dat groter was dan veel grotere computers waarvoor je zo'n tienduizend gulden of meer op tafel moest leggen, terwijl de C-64 ook nog beschikte over een geavanceerde high-resolution optie, prachtige sprites en een unieke geluids-chip.

Al met al een wonder van techniek voor weinig geld. Maar de ontwikkeling staat niet stil en sinds de komst van de 64 is er heel wat veranderd. We beschikken nu over een grotere kennis van alles wat er mogelijk is met een computer en we zijn daardoor aanzienlijk kritischer geworden bij de aanschaf van een nieuw en beter apparaat. Al snel kwam voor de C-64 de confrontatie met soortgelijke machines van andere merken, al of niet voorzien van de C-64 mogelijkheden en aangepast aan latere normen. Commodore probeerde daarop in te spelen, maar bracht het in de tussentijd niet veel verder dan het lanceren van de minder succesvolle C-16, Plus/4, 500 en 700/710 series, waardoor we met onze 64 nogal beschaamd achterbleven bij MSX, Atari en de MacIntosh.

En dan wordt de 128 aangekondigd. Op papier en in ontwerp een technisch mirakel en dat voor zo weinig geld. De grote doorgroeier, de redding voor wie uit zijn 64 jas gegroeid is. Sinds januari staan de bladen er bol van, compleet met zogenaamde tests. Wij hebben ons daar overigens terughoudend in opgesteld, eerst zien, dan geloven, weten we na heel wat jaartjes Commodore computeren. Maar goed, we hebben er nu een kunnen proberen, zij het dan nog zonder de 1571.

Wat blijft er nu uiteindelijk over van alle beloftes en specifikaties die Commodore ons in al die maanden dat we wachtten op de komst van dit wonder heeft voorgeschoteld?

Nederlandse primeur.

Op de onlangs gehouden Commodore-Info beurs in Den Haag hebben we kans gezien CBM zover te krijgen ons een paar machines ter beschikking te stellen. Hoewel Commodore Nederland op deze beurs (wijselijk?) door afwezigheid schitterde, had ieder van de 7.500 bezoekers daardoor toch de gelegenheid de 128 even te kunnen beproeven, of op z'n minst te bekijken op een aantal stands. De machine werd al eerder getoond op de PC-RAI, maar dat was een echt prototype model. De Commodore 'PC' 128 is feitelijk veel meer dan één computer. Het zijn 3 computers in één behuizing: een C-64, zoals we die al lang kennen met een 6510, een nieuwe 128, met alle mogelijkheden van de 64, maar daarnaast ook nog eens 128K RAM, 80 kolommen, een grotere microprocessor, de 8502 en een veel uitgebreidere Basic, versie 7.0. En dan is de 128 ook nog compatible met veel CP/M computers, waardoor hij in staat moet zijn om alle bestaande CP/M software te runnen. En dat alles voor de prijs van zo'n 1300 gulden, hoewel bij het drukken van dit artikel de exacte prijs nog altijd door Commodore geheim wordt gehouden. In de V.S. en enkele Europese landen is er al wat gebeurd met de 128. De V.S. belooft een grote toekomst voor de nieuwe Commodore, waar ze de winkels uitvlogen, in Engeland gaat men ook binnenkort verkopen en West-Duitsland heeft z'n eerste crash met deze machine al beleefd, toen twee weken na de start van de verkoop, alle machines werden teruggenomen en de europese produktie werd stop gezet. En Nederland wacht, nog steeds......

De machines die we hebben getest, zijn in Duitsland vervaardigd voor de Engelse markt, en zijn via België bij ons terecht gekomen, dus een kleine afwijking op de uiteindelijk in Nederland te verkopen versie is niet ondenkbaar. In grote lijnen zal het echter wel dezelfde machine blijven. De tests die we hebben kunnen uitvoeren, zijn gedaan met hulp van de 1541 diskdrive, waardoor onze gegevens qua disksnelheid een vertekend beeld geven ten opzichte van de aangekondigde 1571 drive, die binnenkort geleverd gaat worden.

Het toetsenbord

Vergeleken met de 64, is het toetsenbord van de 128 opmerkelijk groter uitgevoerd, misschien om de indruk te wekken van een krachtiger machine, of is het om intern

ruimte te maken voor de drie processoren met de aangehangen electronica? Ik weet het niet, maar wel is het duidelijk, dat het keyboard veel beter te bespelen is dan alle voorgaande Commodore-toetsenborden, misschien met uitzondering van de PC-10/20. De grote verbetering is het aparte numerieke 'pad', dat rechts van het gebruikelijke QWERTY-bord is geplaatst. Jammer is alleen, dat je op het cijfer- bordje niet direkt de beschikking hebt over alle rekenkundige funktietoetsen, terwijl de cursortoetsen ook niet in een handomdraai te bereiken zijn. De funktietoetsen zijn verplaatst naar de rechterbovenhoek, en horizontaal gerangschikt. Het numerieke 'pad' is wel voorzien van een extra RETURN-toets en een toets voor + en −. Aan de bovenkant van het bord vinden we verder een toets voor 40/80 koloms display, LINE FEED, HELP, CAPS LOCK, ALT, ESC en TAB , met apart nog eens 4 extra cursortoetsen, waarmee we zonder de SHIFT te hoeven gebruiken alle kanten op kunnen. Helaas, geen van deze nieuw aangebrachte toetsen funktioneert in de 64-mode. Uiteraard is dit gebeurd om geen inbreuk te maken op de ons bekende 64 operaties. We zouden daardoor alleen maar in de war gebracht worden, nietwaar. De kans bestaat natuurlijk, dat het aanbrengen van extra software, om deze toetsen voor de 64 geschikt te maken, als resultaat heeft dat een deel van de 64-software niet meer werkt op de 128, en dat is niet de opzet.

Schermdisplay

In de 128-mode wordt door de 40/80 koloms toets bepaald welke scherm-mode wordt gebruikt. Deze toets werkt alleen bij het opstarten van de 128. Is hij ingedrukt, dan wordt aangenomen, dat er met 80 kolommen wordt gewerkt, zo niet, dan blijft het bij 40 kolommmen, zowel voor de 128 als CP/M. De 64-mode kan alleen werken met 40 kolommen, en zal het dan ook niet zondermeer doen op een 80-koloms monitor. Veranderen van 40 naar 80 kolommen en vice versa kan ook, door de (eindelijk) naast de AAN/UIT knop, aan de rechter zijkant aangebrachte RESET-schakelaar in te drukken. Je moet er wel kleine vingers of een pen voor gebruiken, maar het gaat beter dan het uitschakelen van de hele machine. Een probleem is wel, dat je voor 40 kolommen een normale monitor of TV ontvanger kunt gebruiken, zoals je dat gewend bent bij de 64, terwijl voor de 80-karakter display een RGBI-monitor nodig is.

Bij 40 koloms wordt zowel voor de 128 als voor de 64 een composiet videosignaal gebruikt. Dit composiet of samengesteld signaal bevat alle kleurinformatie, waardoor het moeilijk wordt de kleuren te scheiden. RGB(I) heeft de hoofdkleuren ondergebracht in drie aparte signalen, zodat scheiding wel mogelijk is. Voor de 80 koloms vertoning is een aparte video-chip ingebouwd, die kan beschikken over alle 16 kleuren die we kennen van de C-64. Hier en daar vertonen de kleuren wel enige afwijking met de ons bekende tinten. Het aantal beschikbare karakters is verdubbeld, want naast de ons bekende hoofd- en kleine letters kunnen we nu gelijktijdig beschikken over de grafische tekenset. In totaal dus 512 karakters, maar wel alleen met 80 koloms scherm. De chip, die voor de 80 kolommen zorgt, heeft zelf geen hires, graphics of sprites, hoewel je wel een simulatie daarvan kunt maken, op een 48 X 640 scherm. De 80-koloms video gebruikt 16K scherm-RAM, maar dat gaat niet ten koste van de 128K die voor Basic beschikbaar is.

Nog even iets over de verschillen in video-display. Er zitten twee uitgangen op de 128. Één voor 40 en één voor 80 kolommen. Het is echter mogelijk om van beide gelijktijdig gebruik te maken. Hetzij op twee afzonderlijke (kleuren)schermen of op het apart door Commodore ontwikkelde scherm van de (dure) 1901 monitor. Het is een leuke ervaring, om op twee monitoren een verschillend beeld te zien dat uit dezelfde computer afkomstig is.

Maar stel je er ook weer niet teveel van voor, want deze grap heeft ook nadelen, omdat het technisch gezien niet is aan te bevelen, de 80 kolommen te vertonen op een normale monitor of TV ontvanger. Bij de aanschaf van de 128 zit je dus feitelijk ook vast aan een nieuwe monitor. Doe je dat niet, dan kun je, na het aanpassen van de video-uitgang voor 80 kolommen een kwaliteit verwachten, die nog slechter is dan de gemiddelde 80-kolommen kaart voor de 64.

Basic 7.0

Het kon niet anders, of de Basic interpreter van de 128 is uitgebreider dan Basic 2.0 van de C-64. De 128 beschikt over Basic 7.0. Wat het versie nummer inhoudt is niet duidelijk, maar misschien ligt dat besloten in het feit dat 7 twee keer zoveel is als 3.5, de versie van de PLUS/4 en de C-16. Een feit is, dat er een grote serie handige opdrachten en commando's aan Basic is toegevoegd, zodat we nu eindelijk ook voor Commodore kunnen spreken over een volwassen programmeertaal. Al met al krijg je in 128 mode de beschikking over een kleine 200 commando's. Je kunt je waarschijnlijk niet 1,2,3, voor de geest halen waar deze opdrachten allemaal voor kunnen dienen, en de ruimte ontbreekt hier om ze stuk voor stuk te gaan bespreken. Maar als de 128 het succes wordt dat Commodore ervan verwacht, dan zullen we in de komende nummers nog genoeg aandacht aan deze nieuwe Basic besteden.

Om een indruk te geven van Basic 7.0 moet je je voorstellen, dat je begint met de bekende Basic 2.0 uit de 64. Daar voeg je alle denkbare direkte- en programma-opdrachten aan toe, die je kent van de diverse cartridge-tools, terwijl daarnaast nog de commando's van Simons' Basic, Plus/4 en alle bekende Expanders bijkomen. Als klap op de vuurpijl zijn ook alle MSX-commando's, al of niet met een andere naam, in deze Basic interpreter verwerkt. Het geheel is groots van opzet, omdat er ook nog een serie speciaal voor de 128 ontworpen kommando's zijn ingebouwd. Het is duidelijk dat men heeft geprobeerd de gebruiker voor te zijn, door alvast in te spelen op de toekomstige wensen van de 128-bezitters. Dat heeft Commodore in ieder geval geleerd van de C-64.

Ook de communicatie met de disk is niet vergeten. De commando's van Basic 4.0, die worden gebruikt bij de CBM 40xx en 80xx series zijn allemaal overgenomen, zodat ook de drive zonder moeilijke OPEN opdrachten is aan te sturen. Ik denk zelfs, dat het mogelijk moet zijn om de meeste software van de 'business machine' op de 128 te kunnen runnen. Wat denk je b.v. van commando's als **DIRECTORY, SHIFT RUN/STOP** om op te starten, **DSAVE, DLOAD, COLLECT, SCRATCH** en **HEADER**? Stuk voor stuk opdrachten waar we bij de C-64 nogal wat onlogische commando's voor moesten intypen. Ook **DOPEN, DCLOSE** en **RECORD** zijn niet vergeten, zodat het nu niet moeilijk meer is om relatieve bestanden te maken en aan te spreken. En voor dit alles hebben we de ruimte. Maar liefst 128K. Bij het opstarten meldt de 128 dat er meer dan 122.000 Bytes voor Basic beschikbaar zijn. Dat wordt verdeeld in grofweg 64K voor het programma, en net zoveel om de Basic-variabelen op te bergen in het geheugen. De RAM voor Sprites, bitmaps en video zijn hierbij niet inbegrepen.

Tools

De hulpmiddelen om te kunnen programmeren in Basic, die we voor een groot deel terugvinden in de 64-cartridges zitten er ook allemaal in. **AUTO, TRACE, RENUMBER, KEY,** omrekenen van **HEX** naar **DECIMAAL** en terug en noem maar op. Allemaal fantastisch. Ook echt gestructureerde programma's zijn nu te fabriceren, want er zit eindelijk een goede **IF THEN/ELSE** en **DO/WHILE/UNTIL** cyclus in de 128. Naast de bekende AND en OR zit er zelfs een XOR (exclusive OR) ingebouwd. Maar.... helaas heeft dit alles ook z'n keerzijde. Hoewel dit niet direkt te wijten is aan de uitgebreide Basic, is de uitvoering van

deze fantastische Basic programma-set nogal langzaam.

Benchtest

We hebben wat testen gedaan, en die vergeleken met de snelheid waarmee de C-64 dezelfde programma's kan uitvoeren. Terwijl de ingebouwde klok van de 128 twee maal sneller is, blijkt dat de programma's tot ongeveer 30 procent worden vertraagd. En dat is natuurlijk niet zo leuk. Je zou iets anders verwachten van zo'n modern apparaat. De oorzaak van deze vertraging zit waarschijnlijk in de synchronisatie van de drie processoren en de video-chip. Precies is het niet bekend, maar wel weten we, dat ook Commodore hier de grootste problemen mee heeft gehad. Dat zou één van de redenen zijn, waarom de 128 zo lang op zich heeft laten wachten.
Misschien worden de uiteindelijke versies sneller dan de test- exemplaren maar daar is niet veel hoop op. Als pleister op deze wonde heeft Commodore een commando bedacht, dat de prachtige naam **FAST** heeft meegekregen. Daarmee kun je berekeningen e.d. in Basic aanmerkelijk sneller uitvoeren. De tijdwinst is ongeveer 50 procent. Het nadeel bij dit commando is echter, dat de video wordt uitgeschakeld op het moment dat FAST als opdracht wordt gegeven, zodat je niets meer ziet op het scherm. En dat is ook niet alles. Vooralsnog zullen we daarmee moeten leven. Waarschijnlijk komt er wel iemand op een briljant idee, om deze vertraging in het uitvoeren van de programma's op te heffen, want zelfs de uitvoering van machinetaal lijdt onder dit euvel.

CP/M

De CP/M mogelijkheden hebben we alleen kunnen bekijken met hulp van de 1541 diskdrive. En daarin zijn we nogal teleurgesteld. De verwerkingssnelheid van dit operating-systeem is dan zo langzaam, dat je medelijden krijgt met de mensen die daarmee aan het werk moeten. Zo traag als deze combinatie werkt met dit 8-bits systeem is nog niet eerder gesignaleerd. Het is niets beter dan de 64, waarin een CP/M cartridge wordt gestoken. Een teleurstellende ervaring. Voor de mensen die de 128 aanschaffen om er met CP/M op te gaan werken, zit er niets anders op, dan ook de 1571 diskdrive te kopen, want deze werkt een paar maal sneller dan de 1541.

Bovendien is de 1571 in staat om een paar -gangbare- CP/M diskformaten (Osborne/Kaypro) te kunnen lezen, zodat je dan wat software betreft, niet geheel afhankelijk bent van Commodore en bv. de Public Domain CP/M software kunt gaan aanboren. Al met al denk ik, dat voor de liefhebbers van CP/M wel betere mogelijkheden zijn, voor zover MS/DOS geen alternatief oplevert. Je kunt beter de 64 houden voor de bestaande software en daarnaast een goede CP/M of MS-DOS machine kopen, dan alle stukken op één kaart te zetten, door de 128 aan te schaffen. Er is ooit sprake geweest van een 8088 kaart met MS-DOS voor de 128, (zoals die er voor de Amiga is) maar om nu 4 processoren in een machine te combineren is wat al te veel van het goede. Met de huidige drie is het al een tamelijk dure machine om te produceren en dat doet ook vermoeden, dat op de lange duur Commodore wel weer naar iets anders gaat uitkijken voor dit marktsegment.

De 128 beschikt over het CP/M 3.0 operating systeem. De standaard meegeleverde CP/M diskette bevat echter alleen de hoognodige CP/M funkties zoals DIR, PIP, FORMAT en TYPE. De rest van de utilities zal er wel apart bij moeten worden aangeschaft, voorzover ze niet 'off the shelf' kunnen worden betrokken. Het 'BOOTEN' van de CP/M software met de 1541 duurt op zich al een paar minuten. Om de DIRectory van een disk op je scherm te krijgen kost een twintigtal seconden, terwijl ook het laden van de programma's een onaanvaardbaar lange tijd opleverde in onze tests. Bovendien kan de 1541 geen andere CP/M diskformaten lezen. Dat wordt allemaal wel wat beter met het gebruik van de 1571, maar in het algemeen moet je de CP/M mogelijkheden van de 128 toch niet overschatten. Als simulatie van CP/M is het een leuk probeersel, maar die ervaring kennen we ook al van de C-64. Wil je met CP/M aan het werk op de oude 1541, dan zul je moeten wachten tot Commodore de nodige software heeft overgezet op 'Commodore CP/M formaat'.

De 1571 Diskdrive

Het gebruik van de nieuwe 1571 drive zal de mogelijkheden en de snelheid waarmee met de 128 gewerkt kan worden, aanzienlijk opvoeren. Als je de 1571 in 64-mode gebruikt, is hij identiek aan de 1541, hoewel de meeste Turbo-utilities niet zullen werken. Ook de kopie-beveiligde 1541 software kan niet runnen met een 1571 drive. Maar volgens Commodore komt de 128 pas goed tot z'n recht als deze drive wordt aangesloten en er zijn inderdaad nogal wat goede eigenschappen op te noemen van de 1571.
De geheugenkapaciteit van de 1571 is verdubbeld t.o.v. de 1541. Dit komt omdat er nu door twee koppen aan beide zijden van de disk wordt geschreven. Dat betekent zo'n 360K tegenover de 170K voor de 1541. In CP/M gaat er zelfs 410K op dezelfde schijf. Je moet er wel om denken, dat het bij de 1571 wenselijk is om op z'n minst dubbelzijdige (DSDD) disks te gebruiken, die vrij prijzig zijn. De snelheid waarmee de 1571 werkt is vijf tot tien maal groter dan wat we gewend zijn van de oude 1541. Kortom er zit nogal wat verschil tussen die twee. En dat mag ook wel, want om eerlijk te zijn is de 1541 niet zo'n verbluffend goed apparaat, dus zo verwonderlijk is het niet, dat de 1571 beter uit de bus komt. In Parijs zagen we overigens al weer een enkelzijdige uitvoering, de 1570, die in de 1541 behuizing zit en iets goedkoper gaat worden.

Dokumentatie

Naast een klein en kleurrijk instruktieboekje, waarin alle gegevens om snel aan het werk te kunnen bondig zijn opgenomen, krijg je bij aankoop van de 128 tevens een Engelstalige System Guide in handen.
Dit - meer dan 400 pagina's tellende - boekwerk geeft gedetailleerde informatie over de drie afzonderlijke modes die in de 128 zijn ingebouwd. Alle Basic commando's worden uitvoerig omschreven en waar nodig verduidelijkt met programma voorbeelden.
Helaas zijn er in het hele boek maar een paar bladzijden aan de CP/M-kant van de 128 gewijd. Misschien komt dit omdat Commodore zelf ook van mening is, dat de CP/M-mode niet de meest succesvolle mogelijkheid is van de 128. Voor uitvoerige gegevens over CP/M wordt simpelweg verwezen naar de algemene 'CP/M Users Guide', die weliswaar overal te koop is, maar niets met Commodore te maken heeft. Verder alle lof over de meegeleverde literatuur.
In het nederlands is er inmiddels het Premiereboek voor de Commodore 128 van Data Becker verschenen. Dit boek kost zo'n 50 gulden en geeft een leesbare samenvatting van de gegevens die je terug kunt vinden in het engelse manual. Maar de korte tijd waarin Data Becker dit boekje moest samenstellen heeft jammergenoeg zijn effect in de nodige slordigheden die er in voorkomen. Zo kan het gebeuren, dat we in het hoofdstuk over grafische commando's alles kunnen lezen over de opdrachten waarmee we de drive kunnen besturen. Zo zijn er nog wel een paar oneffenheden op te noemen.
Voor de start met de 128 echter een aardig boek om er in het nederlands bij te hebben, temeer daar de vertaling van de engelse handleiding nog wel even op zich zal laten wachten.
Wil je dieper ingaan op de mogelijkheden van de 128, dan ben je vooralsnog aangewezen op de 2 duitse uitgaven van Data Becker over de 128, die werkelijk alles blootgeven van wat er binnenin de 128 schuilt. Een compliment voor deze prestatie, terwijl de computer nog niet eens op de markt is, tenminste als het allemaal klopt. Waarschijnlijk worden deze boeken snel vertaald.

Waar voor je geld?

Ik ben me ervan bewust dat er in deze opsomming nogal wat goede en misschien ook minder goede mogelijkheden van de 128 onbesproken zijn gebleven. Maar toch geloof ik, dat er een redelijk beeld is gegeven van de eerste indruk van het werken met een echte 128. Het is helaas niet allemaal zo mooi als de aankondigingen doen voorkomen. We hebben lang op deze machine moeten wachten, misschien wel te lang. Want in het tussenliggende jaar zijn er een aantal computers van concurrerende merken op de markt gekomen, die in het algemeen niet onderdoen voor de Commodore PC 128. Een van die machines bevindt zich in de eigen Commodore omgeving en is de PC-10/20. Laten we ook de Amiga (rond de 8 mille in standaarduitvoering) niet vergeten, maar die zal nog wel even op zich laten wachten. De prijs/prestatieverhouding van de Atari 520 ST ligt wat gunstiger. Dit komt vooral door de randapparaten bij de 128, wanneer je tenminste alle eigenschappen van deze computer voldoende tot hun recht wil laten komen. Dan ben je voor een compleet 128 systeem toch snel zo'n 4500 gulden kwijt. En voor dat geld zijn er op het moment een hele serie andere computers te koop, die op een paar uitzonderingen na, dezelfde eigenschappen hebben als de zo lang verbeide 128.
Voor hen die inmiddels zijn uitgekeken op de 64 zal er niet veel anders opzitten, dan serieus alle mogelijkheden en wensen op een rij te zetten. Pas dan kunnen ze inschatten, of de 128 wel echt de computer is, die hierop een antwoord geeft. ●

Jan Bodzinga.

CP/M

De Control Program for Microprocessors (CP/M) 3.0 Commando's voor de derde bedrijfstoestand (mode), waarbij de aparte Z80 met 4 MHz kloksnelheid benut wordt, zijn naast een aantal kerncommando's zoals TYPE of FORMAT:
DATE ★ CCP ★ DEVICE ★ DIR ★ DUMP ★ ERASE ★ GENCOM ★ GET ★ PUT ★ SAVE ★ PIP ★ CPM ★ INITDIR ★ SET ★ CURSOR ★ ★ ★

```
250 ifa=3 then poke 1125,z
260 ifa=4 then poke 1239,z
270 ifa=5 then poke 1242,z
280 ifa=6 then poke 1245,z
290 ifa=7 then poke 1359,z
300 ifa=8 then poke 1362,z
310 ifa=9 then poke 1365,z
315 y=y+1
320 ifpeek(1119)=z  and   peek(11
    22)=z  and   peek(1125)=z  then
      goto 640
330 ifpeek(1239)=z  and   peek(12
    42)=z  and   peek(1245)=z  then
      goto 640
340 ifpeek(1359)=z  and   peek(13
    62)=z  and   peek(1365)=z  then
      goto 640
350 ifpeek(1119)=z  and   peek(12
    39)=z  and   peek(1359)=z  then
      goto 640
360 ifpeek(1122)=z  and   peek(12
    42)=z  and   peek(1362)=z  then
      goto 640
370 ifpeek(1125)=z  and   peek(12
    45)=z  and   peek(1365)=z  then
      goto 640
380 ifpeek(1119)=z  and   peek(12
    42)=z  and   peek(1365)=z  then
      goto 640
390 ifpeek(1125)=z  and   peek(12
    42)=z  and   peek(1359)=z  then
      goto 640
400 if y=9 then goto 470
410 if s=1 then goto 200
420 if s=2 then goto 180
430 printtab(6)"[2xneer]gewonnen
    door: speler (1)":p1=p1+1
440 goto 490
450 printtab(6)"[2xneer]gewonnen
    do0r: speler (2)":p2=p2+1
460 goto 490
470 printtab(16)"[neer]remise"
490 printtab(9)"[2xneer]nog een k
    eer (j/n)"
500 getq$:ifq$=""then 500
510 ifq$="j" then goto 20
520 printchr$(147)
530 printtab(14)"[2xneer]einduits
    lag"
540 fork=1198to1208:pokek,67:next
560 printtab(13)"[3xneer]speler (
    1):";p1-48
580 printtab(13)"[neer]speler (2)
    :";p2-48
590 getz$:ifz$=""then590
600 end
640 if z=86 then goto 430
650 if z=87 then goto 450

         ** EINDE LISTING ***
```

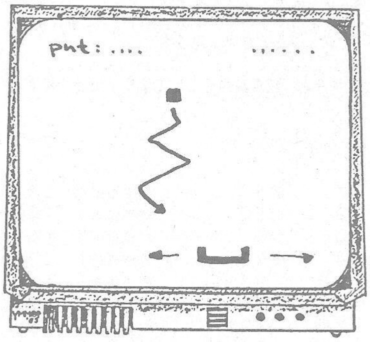
boter kaas & eieren

regel 1	127	regel 65	254
regel 2	16	regel 70	121
regel 3	34	regel 75	43
regel 4	143	regel 80	58
regel 5	143	regel 85	137
regel 10	121	regel 90	252
regel 20	8	regel 100	80
regel 30	88	regel 110	89
regel 40	77	regel 120	107
regel 50	47	regel 130	95
regel 55	48	regel 140	43
regel 60	90	regel 150	50

regel 155	145	regel 380	116
regel 160	201	regel 390	116
regel 170	135	regel 400	154
regel 180	238	regel 410	131
regel 190	66	regel 420	139
regel 200	128	regel 430	230
regel 210	68	regel 440	38
regel 230	63	regel 450	202
regel 240	58	regel 460	38
regel 250	62	regel 470	230
regel 260	69	regel 490	255
regel 270	64	regel 500	130
regel 280	68	regel 510	210
regel 290	75	regel 520	77
regel 300	70	regel 530	105
regel 310	74	regel 540	49
regel 315	63	regel 560	154
regel 320	107	regel 580	122
regel 330	116	regel 590	157
regel 340	125	regel 600	128
regel 350	125	regel 640	204
regel 360	107	regel 650	207
regel 370	116	ready.	

Catch the Ball 64

Dit eenvoudige en korte spelletje lijkt heel simpel te spelen, maar in de data-regels aan het eind zitten een paar slimme routines, die de beweging van het balletje en van het mandje in toenemende mate in de war sturen. Werkt met een joystick in poort 2. Complimenten voor de zeer hoge resultaat/regel verhouding.



```
  1 rem catch the ball / c-64
  2 rem door arnoud van de wal
  3 rem uit apeldoorn / 055-66225
    8
  4 rem
 10 ifpeek(49152)=120then60
 20 fort=832to960:poket,0:next
 30 fort=862to880step3:poket+2,31
    :poket,248:poket+34,255:next
 40 fort=883to895:poket,255:next
 50 fort=49152tot+61:reada:poket,
    a:next
 60 c=53248:poke2040,13:poke2041,
    14:pokec+39,7:pokec+40,1:poke
    c,150:pokec+32,6
 70 pokec+1,180:poke254,0:print"[CLR-
    HOME][GROEN]";chr$(08)chr$(14
    2)
 80 b$="[8xrechts]":a$=b$+"[RVS-aan]
    [22xspatie]":c$="[HOME][4xneer]"
 90 printc$a$:fort=0to13:printb$"
    [RVS-aan] "tab(29)" ":next:print
    a$
100 sys49152:poke767,0:pokec+21,3
```

```
110 printc$tab(20)"[WIT]QQQQQ":print
    "[HOME]"b$"[3xspatie]catch th
    e ball
120 printb$"[neer] score[4xspatie]
    balls left":l=5
130 x=int(rnd(0)*130)+100:y=93
140 fort=0to4:r=int(rnd(0)*3)-1:r
    =r*s/50
150 foru=0to19:x=x+r:y=y+1:ifx>23
    0orx<100thenr=-r
160 pokec+2,x:pokec+3,y:next:next
170 a=peek(c):ifx+5>aanda>x-20then
    s=s+10:goto200
180 fort=0to01:pokec+17,24:q=2↑20
    :pokec+17,31:q=2↑10:next:poke
    c+17,27
190 l=l-1:printc$tab(24-l)"[GROEN]
    [RVS-aan]V":ifl<.then220
200 printc$"[WIT]"tab(10);s:ifs>1
    20thenpoke254,1
210 pokec+3,1:fort=0to500:next:goto
    130
220 printc$"[5xneer]"b$b$"[2xlinks]
    [WIT]game over[l.blauw]":poke
    c+21,0
230 get x$:if x$="" then 230
240 run
250 data120,169,13,141,20,3,169,1
    92,141,21,3,88,96,173,255,2,2
    4,109,0,208
260 data201,97,176,2,169,97,201,2
    31,144,2,169,231,141,,208,162
    ,254,173,,220
270 data168,41,4,240,11,162,2,152
    ,41,8,240,4,166,254,208,3,142
    ,255,2,76,52,234

              *** EINDE LISTING ***
```

```
catch the ball

   regel 1     247       regel 140    107
   regel 2     222       regel 150    46
   regel 3     88        regel 160    40
   regel 4     143       regel 170    156
   regel 10      245     regel 180    158
   regel 20      164     regel 190    0
   regel 30      129     regel 200    150
   regel 40      29      regel 210    81
   regel 50      229     regel 220    85
   regel 60      178     regel 230    144
   regel 70      116     regel 240    138
   regel 80      233     regel 250    3
   regel 90      92      regel 260    103
   regel 100      244    regel 270    240
   regel 110      157
   regel 120      214
   regel 130      120    ready.
```

Faculteit

Een rekengrapje, waarbij de floating point routines niet worden gebruikt en toch vrij snel het resultaat (getallen tot in de miljarden) op het beeldscherm komt.

```
 1 rem faculteit
 2 rem door c de gelder
 3 rem berkel en roderijs / 0189
   1-3130
 4 rem
10 dima%(2000):r=1:a%(1)=1:forv=
   2to1000:l=r:c=0:fort=1tol:a=a
   %(t)*v
20 b=int((a+c)/1000):a%(t)=a-b*1
   000+c:c=b:r=l:ift=landc>0then
   r=r+1:a%(r)=c
30 nextt:printv"!= ";:n$="":fory
   =rto1step-1:g=a%(y):f=len(str$
   (g))-1
40 printleft$(n$,3-f)right$(str$
   (g),f);:n$="000":next:print:next

              EINDE LISTING ***
```

```
faculteit

   regel 1    48       regel 20    91
   regel 2    66       regel 30    177
   regel 3    95       regel 40    20
   regel 4    143
   regel 10    210     ready.
```

Beeldeffect

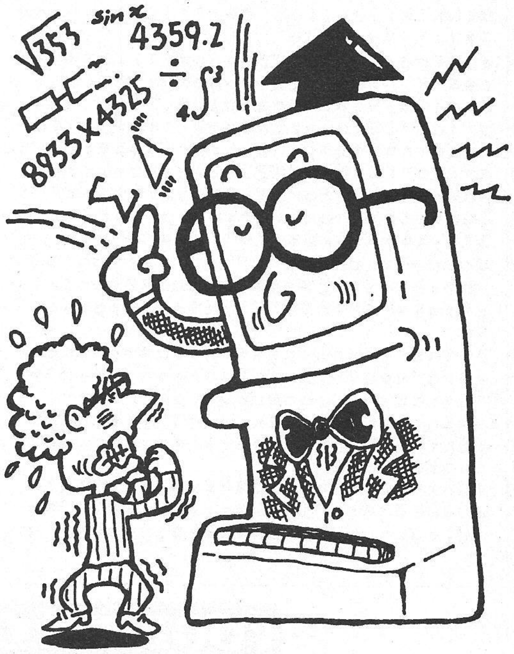
In twee regels een lopend rasterpatroon op het scherm, waar je nog de tekst doorheen kunt lezen.

```
 1 rem beeldeffect
 2 rem door johan van toledo
 3 rem zaandam / 075 350187
 4 rem
10 print"[CLR-HOME]":forn=0to9:read
   y:poke828+n,y:next
20 sys828
30 data120,206,33,208,123,33,208
   ,76,61,3
        * EINDE LISTING ***
```

regel 1	152	regel 10	254
regel 2	223	regel 20	64
regel 3	142	regel 30	215
regel 4	143		

Tafels

Een rekenspelletje voor de kinderen, waarmee ze de tafels van vermenigvuldiging kunnen oefenen, sommetjes maken en laten overhoren door de computer. Erg kort, maar toch effectief en met weglating van regel 10, die de schermkleuren pookt, ook te gebruiken voor andere CBM computers.




```
  1 rem tafels-64
  2 rem door ad van pinxteren
  3 rem uit eemnes / 02153-14631
  4 rem
 10 poke53280,6:poke53281,6
 20 print"[CLR-HOME][3xneer][6xspatie
    ][ROOD]tafels 64":print"[neer]
    [6xspatie][WIT]by adsoft":print
    "[neer][6xspatie][1.blauw]hoo
    fdmenu
 30 print"[3xneer][5xspatie]1.[2xspat
    ie]tafels bekijken":print"[3xneer
    ][5xspatie]2.[2xspatie]tafels
     overhoren"
 40 print"[3xneer][5xspatie]3.[2xspat
    ie]stoppen":print"[3xneer][3xspat
    ie]kies een nummer ";
 50 inputa$
 60 ifa$="1"then320
 70 ifa$="2"then100
 80 ifa$="3"then print"[CLR-HOME]"
    :end
 90 goto50
100 print"[CLR-HOME][8xspatie][1.blau
    w]overhoring":print"[2xneer][5xsp
    atie]submenu"
110 print"[2xneer][5xspatie]1.[2xspat
    ie]tafels apart":print"[2xneer]
    [5xspatie]2.[2xspatie]tafels
    algemeen"
120 print"[3xneer]kies een nummer
     :";:inputa$
130 ifa$="1"then240
140 ifa$="2"then160
150 goto120
160 print"[2xneer]hoeveel sommen"
    ;:inpuths:print"[neer]tot hoe
    ver mag ik gaan";:inputhv
170 print"[CLR-HOME]":g=0:forx=1
    to hs:a=int(rnd(0)*hv)+1:b=int
    (rnd(0)*10)+1
180 print x".[2xspatie]";a;"V";b;
    "=";:inputan
190 if an=a*bthengosub360:goto210
200 if an<>a*bthengosub370
210 nextx
220 print g;" van de";hs;"goed.":
    ifg=hsthenprint"[neer] helema
    al goed, dus een [ROOD]10[1.blauw
    ] !"
230 gosub380:goto20
240 print"[2xneer]hoeveel sommen"
    ;:inpuths:print"[neer]welke t
    afel";:inputhv
250 print"[CLR-HOME]":g=0:forx=1
    to hs:a=int(rnd(0)*10)+1
260 print x".[2xspatie]";a;"V";hv
    ;"=";:inputan
270 ifan=a*hvthengosub360:goto290
280 ifan<>a*hvthengosub370
290 nextx
300 print g;" van de";hs;"goed.":
    ifg=hsthenprint"[neer] helema
    al goed, dus een [ROOD]10[1.blauw
    ] !
310 gosub380:goto20
320 print"[CLR-HOME][6xspatie]taf
    els bekijken"
330 print"[3xneer]welke tafel wil
     je zien";:input wt
340 forx=1to10:print x;"V";wt;"="
    ;x*wt
350 nextx:gosub380:goto20
360 print"[op][30xrechts][WIT]goe
    d ![1.blauw]":g=g+1:return
370 print"[op][30xrechts][ZWART]f
    out ![1.blauw]":return
380 print"[2xneer]druk <space> vo
    or hoofdmenu"
390 gets$:ifs$=""then390
400 return

        *** EINDE LISTING ***
```

tafels

regel 1	229	regel 150	28
regel 2	234	regel 160	107
regel 3	148	regel 170	244
regel 4	143	regel 180	246
regel 10	49	regel 190	30
regel 20	182	regel 200	123
regel 30	146	regel 210	218
regel 40	247	regel 220	221
regel 50	234	regel 230	77
regel 60	83	regel 240	15
regel 70	80	regel 250	95
regel 80	234	regel 260	82
regel 90	238	regel 270	130
regel 100	101	regel 280	215
regel 110	43	regel 290	218
regel 120	129	regel 300	187
regel 130	84	regel 310	77
regel 140	86	regel 320	114

```
regel 330    74     regel 380    6
regel 340    205    regel 390    141
regel 350    97     regel 400    142
regel 360    208
regel 370    37     ready.
```

Ketting 64

Bordspelletjes, waarmee je tegen de computer kunt spelen in een poging om alle stenen op een rij te krijgen, waarbij er een toevalseffect wordt ingebouwd via een electronisch dobbelsteen in regel 360, waar een integer (1 tot 6) via RND wordt bepaald. In het begin is wat uitleg over de spelregels. De kleurkeuze is met grijs en rood en wit/zwarte stenen aangenaam voor het oog. Weer een kort programma met toch een heleboel inhoud.

```
  1 rem ketting
  2 rem door t veugen
  3 rem delft / 015-619128
  4 rem
 10 n=9:t=10:f=11:u=27:fori=1to5:
    w(i)=0:w(f-i)=f:b(i)=f:b(f-i)
    =0:next:e6=1106
 20 poke53280,15:poke53281,15:poke
    53272,21:w$="[WIT]Q":print"[CLR-H
    OME][neer][BLAUW][2xspatie]u
    speelt met wit '[WIT]Q[BLAUW]
    '"
 30 z$="[ZWART]*":print"[neer][2xspat
    ie]c64 speelt met zwart '[ZWART]
    *[BLAUW]'[neer]":deffne(x)=k*
    (f-k)/t*(f-l)/t/x
 40 print"[2xspatie]doel spel : e
    en aaneengesloten [ROOD]ketti
    ng[BLAUW]"spc(14)"te vormen."
    :print"[neer][2xspatie]u zet
    ve";
 50 print"rticaal, c64 zet horizo
    ntaal.":print"[neer][2xspatie]
    ingesloten schakels worden te
    ruggezet."
 60 deffnp(x)=x*(f-x)=0:print"[2xneer
    ][2xspatie]succes !":print"[2xnee
    r][2xspatie]wilt u beginnen (
    j/n) ?"
 70 a1=65:a2=74:n1=48:n2=57:h2=20
    0:wait198,1:getf$:iff$<>"j"and
    f$<>"n"goto70
 80 deffnd(x)=abs(x-sgn(x)):h$="|"
    :s$=" ":e$="c64":gosub100:on(
    asc(f$)-70)/4goto150,210
 90 print"[3xspatie]"a$;:forj=1to
    n:printb$c$;:next:printb$;:return
100 print"[CLR-HOME][3xspatie]";:
    poke646,2:fori=a1toa2:print" "
    chr$(i);:next:print:print"[2xneer
    ]";:a$="┌":b$="─":c$="┬"
110 gosub90:print"┐":fori=1ton:a$
    =h$:b$=s$:c$=a$:gosub90:print
    a$:a$="├":b$="C":c$="┼":gosub
    90
120 print"┤":next:a$=h$:b$=s$:c$=
    a$:gosub90:printa$:a$="└":b$=
    "C":c$="┴":gosub90:print"┘"
130 print"[HOME][4xneer]";:fori=0
    ton:printchr$(48+i):print:next
    :poke646,7
140 fork=1tot:x=2+2*k:y=2+2*w(k):
    gosub370:printw$;:y=x:x=2+2*b
    (k):gosub370:printz$;:next:return
```

```
150 gosub360
160 x=u:y=5:gosub370:print"uw zet
     ? ..[2xlinks]";:g1=a1:g2=a2:
    gosub380:k=g:g1=n1:g2=n2:gosub
    380
170 l=g:if(k<0ork>t)or(l<0orl>t)or
    peek(e6+2*k+80*l)<>32goto160
180 v=0:m1=0:m2=0:ifabs(w(k)-l)<>
    rgoto160
190 w(k)=l:fori=1ton:v=v+fnd(w(i)
    -w(i+1)):next:gosub300:b(m1)=
    -f*(k<7):b(m2)=-f*(k<5)
200 gosub100:q=0:fori=1tot:q=q-fn
    p(w(i)):next:q=q=0:ifv=0andqthen
    v$="u":goto400
210 gosub360:em=0:mm=0:x=u:y=5:gosub
    370:printe$" denkt[9xlinks][2xnee
    r]en zet ";:v$=e$
220 forl=1tot:kk=b(l):ke=kk*(f-kk
    ):k=kk-r:gosub260:k=kk+r:gosub
    260:b(l)=kk:next:k=km:l=lm
230 p=e6+2*k+80*l:b(l)=k:m3=0:m4=
    0:gosub330:w(m3)=-f*(k<7):w(m
    4)=-f*(k<5)
240 print"[grijs 1]"chr$(k+64)chr$
    (l+47);:fori=1to5:forj=1toh2:
    next:pokep,42:forj=1toh2:next
    :pokep,32:next
250 gosub100:onint(em/h2)+1goto15
    0,400
260 b(l)=k:if(k<1ork>t)orpeek(e6+
    2*k+80*l)-32thenreturn
270 c=0:q=0:fori=1ton:c=c+fnd(b(i
    )-b(i+1)):next:fori=1tot:q=q-
    fnp(b(i)):next:q=q=0
280 ev=90-c-(c=0andq)*110-n*(ke=0
    )+fne(3):ifev>emthenem=ev:km=
    k:lm=l:ifem>=h2goto240
290 m3=0:m4=0:gosub330:i=m3+m4>0:
    em=em-i*(100-em):km=km-i*(k-k
    m):lm=lm-i*(l-lm):return
300 ifk>2thene=w(k-2):d=(e-l)/2:if
    d=sgn(d)ande>0ande<fthenifb(l
    +d)=k-1thenm1=l+d
310 ifk<nthene=w(k+2):d=(e-l)/2:if
    d=sgn(d)ande>0ande<fthenifb(l
    +d)=k+1thenm2=l+d
320 return
330 ifl>2thene=b(l-2):d=(e-k)/2:if
    d=sgn(d)ande>0ande<fthenifw(k
    +d)=l-1thenm3=k+d
340 ifl<nthene=b(l+2):d=(e-k)/2:if
    d=sgn(d)ande>0ande<fthenifw(k
    +d)=l+1thenm4=k+d
350 return
360 poke646,6:x=u:y=2:gosub370:r=
    1+int(6*rnd(1)):print"aantal[6xli
    nks][neer]ogen :"r:return
370 poke211,x:poke214,y:sys58732:
    return
380 poke198,0:wait198,1:geta$:g=asc
    (a$)-g1+1:ifg<1org>g2-g1+1then
    380
390 printa$;:return:rem
              +-+-+- tis veugen, de
    lft, 29-6-85 -+-+-+
400 x=u:y=12:gosub370:print"[ROOD]"
    v$" wint !":x=0:y=24:gosub370
    :wait198,1


          *** EINDE LISTING ***
```

checksum : KETTING

regel	1	165	regel	200	198
regel	2	225	regel	210	43
regel	3	43	regel	220	173
regel	4	143	regel	230	115
regel	10	219	regel	240	4
regel	20	142	regel	250	98
regel	30	96	regel	260	129
regel	40	253	regel	270	33
regel	50	147	regel	280	159
regel	60	241	regel	290	238
regel	70	48	regel	300	114
regel	80	109	regel	310	143
regel	90	42	regel	320	142
regel	100	83	regel	330	116
regel	110	117	regel	340	145
regel	120	217	regel	350	142
regel	130	244	regel	360	76
regel	140	182	regel	370	69
regel	150	38	regel	380	158
regel	160	184	regel	390	195
regel	170	190	regel	400	7
regel	180	249			
regel	190	3	ready.		

Machtsverheffen

Ook weer rekenen, maar nu is het resultaat X tot de Y-de macht.

```
 1 rem machtsverheffen
 2 rem door c de gelder
 3 rem berkel en rodenrijs / 018
   91-3130
 4 rem
10 input"x↑y x=";x:input"[4xspatie]
   y=";y
20 dima%(2000):r=1:a%(1)=1:forv=
   1toy:l=r:c=0:fort=1tol:a=a%(t
   )*x
30 b=int((a+c)/1000):a%(t)=a-b*1
   000+c:c=b:r=l:ift=landc>0then
   r=r+1:a%(r)=c
```

```
 40 next:printx"↑"v"= ";:n$="":for
    y=rto1step-1:g=a%(y):f=len(str$
    (g))-1
 50 printleft$(n$,3-f)right$(str$
    (g),f);:n$="000":next:print:next
    :run


            *** EINDE LISTING ***
```

machtsverheffen

regel 1	232	regel 20	107
regel 2	66	regel 30	91
regel 3	173	regel 40	54
regel 4	143	regel 50	216
regel 10	45	ready.	

Blauwe Bonen

Schiet maar raak, maar wel met de A en L toets.

```
  1 rem blauwe bonen / cbm-64
  2 rem door hans van leeuwen
  3 rem uit akersloot
  4 rem
 10 rem u kunt met de 'a' en 'l'
    toets sturen en met de spatie
    balk vuren.
 20 poke53281,7:poke53280,0:poke6
    46,6
 30 input"[CLR-HOME] geef level (
    1-9)";le
 40 print"[CLR-HOME]":vl=1:v=5429
    6:w=54276:a=54277:h=54273:l=5
    4272:li=3:i=1
 50 a$(1)="├":a$(2)="[RVS-aan]▒":
    a$(3)="┬":a$(4)="[RVS-aan]▒"
 60 a$(5)="┤":a$(6)="▒":a$(7)="┴"
    :a$(8)="▒"
 70 forx=1to8:readbc(x),st(x):next
 80 print"[3xneer][GROEN]score=";
    pu:print"lives=";li:print"lev
    el=";le:printchr$(19)
 90 ifli=0thenend
100 ifpu=100*ithenle=le-1:i=i+1
110 ri=int(rnd(1)*8)+1
120 forcs=bc(ri)to1524step-st(ri)
130 fory=1tole
140 print"[20xrechts][11xneer][ROOD]"
    a$(vl)chr$(19):geta$
150 ifa$="l"thenvl=vl+1
160 ifa$="a"thenvl=vl-1
170 ifa$=" "then230
180 ifvl>8thenvl=1
190 ifvl<1thenvl=8
200 nexty:pokecs,42
210 ifcs=1524thenli=li-1:goto310
220 nextcs:goto110
230 forx=15to0step-1:pokev,x:poke
    w,129:pokea,15:pokeh,40:pokel
    ,200:next
240 pokew,0:pokea,0
250 forsc=1524+st(vl)tobc(vl)step
    st(vl):pokesc,81
260 ifsc=csthenpu=pu+10:goto280
270 next
280 forx=1524+st(vl)tobc(vl)steps
    t(vl):pokex,32:next
290 ifsc<>csthen180
300 goto80
310 forx=bc(ri)tocsstep-st(ri):poke
    x,32:next:goto80
320 data 1543,1,2016,41,2004,40,1
    992,39,1504,-1,1032,-41,1044,
    -40,1056,-39
```

regel 1	89	regel 150	170
regel 2	231	regel 160	160
regel 3	53	regel 170	34
regel 4	143	regel 180	66
regel 10	97	regel 190	68
regel 20	255	regel 200	212
regel 30	159	regel 210	85
regel 40	139	regel 220	109
regel 50	49	regel 230	201
regel 60	36	regel 240	184
regel 70	191	regel 250	52
regel 80	91	regel 260	116
regel 90	41	regel 270	130
regel 100	24	regel 280	112
regel 110	79	regel 290	91
regel 120	145	regel 300	241
regel 130	242	regel 310	190
regel 140	18	regel 320	237

Geldweg

Programma voor de C-16



```
1 rem c16-geldweg
2 rem door eddie van hemert
3 rem uit ammerzoden / 04199-2589
4 rem
10 vol7
30 print""
35 print"   het spel gokken met de compu
ter"
36 print" wordt gespeeld met 2 'rollen'.
"
37 print" eerst moet je je beginkapitaal
"
38 print" opgeven. en daarna wordt er ge
vraagd"
40 print" hoeveel je inzet. daarna probe
er"
42 print" je het. als je een score van 2
,3"
```

```
44 print" of 12 hebt in het begin dan "
46 print" win je. als je een score"
48 print" van 7 of 11 hebt dan verlies j
e."
50 print" als je een andere score dan"
52 print" 2,3,7,11 of 12 hebt dan moet j
e"
54 print" dat getal wat je dan hebt hale
n."
56 print" als je dat getal dan hebt dan
"
58 print" win je,maar als je dan 7 of 11
"
60 print" hebt verlies je.
62 print" je moet net zolang doorgaan "
64 print" totdat je een van die drie (7,
11
66 print" of het getal om te winnen)
67 print" haalt.
68 print
70 print" druk een toets om verder te ga
an.
72 geta$
74 if a$=""then 72
76 rem ***********************
78 rem *de 'rollen' opzetten*
80 rem ***********************
82 a$="          U-------I    U-------I"
84 b$="          I       I    I       I"
86 c$="          J-------K    J-------K"
88 rem ************************
90 rem *het verkrijgen van een*
92 rem *begin kapitaal        *
94 rem ************************
96 print""
98 input"   begin kapitaal";c
100 rem *********************
102 rem *het starten van de *
104 rem *volgende weddeschap*
106 rem *********************
108 print"   druk een toets voor
           volgende weddeschap"
110 get r$
112 if r$=""then 110
114 print"   je kapitaal is nu";c
116 input"   hoeveel zet je in";w
118 if w > 0 then 124
120 print"   doe niet zo stom!"
122 goto 108
124 if w<=c then 136
126 print"   zoveel heb je niet!"
128 goto108
130 rem *************************
132 rem *organiseer het proberen*
134 rem *************************
136 print"   eerste keer (inzet=";w;")"
138 print"";a$
140 forj=1to5
142 printb$
144 nextj
146 printc$
148 rem ***********************
150 rem *de cijfers verkrijgen*
152 rem ***********************
154 q=int(10+50*rnd(0))
156 forz=1toq
158 a=int(1+6*rnd(0))
160 b=int(1+6*rnd(0))
162 rem ******************************
164 rem *het geluid bij het draaien *
166 rem *van de getallen voor de gok*
168 rem ******************************
170 sound1,700+3*(a*a+b*b),4
172 print"";a;"";b
174 next z
```

```
176 rem ********************************
178 rem *het optellen van de getallen*
180 rem ********************************
182 t=a+b
184 rem ****************
186 rem *overspringen als*
188 rem *de speler wint  *
190 rem ****************
192 if t=7 then 290
194 if t=11 then 290
196 rem ********************
198 rem *overspringen als de*
200 rem *speler verliest    *
202 rem ********************
204 if t=2 then 338
206 if t=3 then 338
208 if t=12 then 338
210 print
212 print
214 print
216 print"    je moet";t;"krijgen voor 7"
218 print"    druk een toets om verder te
 gaan"
220 get r$
222 if r$ = "" then 220
224 print"    volgende kans(inzet=";w;")"
226 print"    proberen om";t;"te krijgen"
228 print""
230 print a$
232 forj=1to5
234 print b$
236 next j
238 print c$
240 rem ***********************
242 rem *laat de getallen zien*
244 rem ***********************
246 q=int(10+10*rnd(0))
248 for z=1 to q
250 a=int(1+6*rnd(0))
252 b=int(1+6*rnd(0))
254 sound1,700+3*(a*a+b*b),4
256 print"";a;"";b
258 next z
260 rem *************************
262 rem *als a+b=t:despeler wint*
264 rem *************************
266 if a+b=t then 290
268 rem ***************************
270 rem *als a+b=7:speler verliest*
272 rem ***************************
274 if a+b=7 then 338
276 rem **********************************
278 rem *anders doet de speler nog eens*
280 rem **********************************
282 goto218
284 rem ***********
286 rem *speler wint*
288 rem ***********
290 print"   je wint"
292 rem ***************************
294 rem *optellen van het gewonnen*
296 rem *         bedrag         *
298 rem ***************************
300 c=c+w
302 rem ********************
304 rem *het winnende geluid*
306 rem ********************
308 forj=1to500:nextj
310 sound1,834,32
312 sound1,798,24
314 sound1,810,8
316 sound1,834,32
318 sound1,739,32
320 sound1,770,8
322 sound1,798,8
```

```
324 sound1,810,8
326 sound1,834,8
328 sound1,810,16
330 sound1,798,16
332 sound1,770,64
334 goto108
336 rem ****************
338 rem *speler verliest*
340 rem ****************
342 print"   je verliest"
344 rem ***********************
346 rem *het verliezende geluid*
348 rem ************************
350 for j=1to500:nextj
352 for x=800to1000 step 4
354 sound1,x,1
356 sound1,x+23,1
358 nextx
360 rem ***********************
362 rem *het aftrekken van het*
364 rem *  verloren bedrag   *
366 rem ***********************
368 c=c-w
370 ifc=0then374
372 if c > 0 then 108
374 if ed= 2then 380
376 print"voor deze keer krijg je 100 pu
nten":c=100
378 ed=2   :goto108
380 print"je bent gebroken "
382 fored=1to1000:next
384 printchr$(147)" commodore basic v3.5
 12277 bytes free"
386 end
```

Vic Liststop

Om even de listing te kunnen stopzetten met toets Shift/shiftlock. VIC-20. Zie remregels.

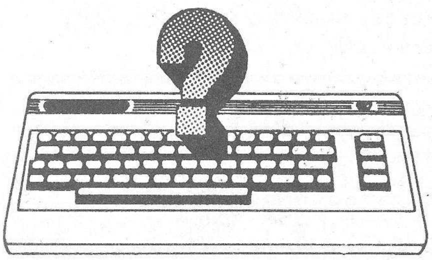
```
 1 rem liststop / vic 20
 2 rem door jacco de leeuw
 3 rem uit zaandam
 4 rem
 5 rem listing stoppen d.m.v. sh
   ift & shift lock
 6 rem prg uit d.m.v. poke774,26
   :poke775,199
 7 rem prg inschakelen d.m.v. po
   ke 774,188:poke775,2
10 fori=700to728:readj:t=t+j:poke
   i,j:next
20 ift<>4054thenprint"[CLR-HOME]
   fout in data":end
30 data133,251,169,0,133,198,165
   ,197,201,62
40 data208,5,169,147,32,210,255,
   173,141,2
50 data41,3,208,234,165,251,76,2
   6,199
60 poke774,188:poke775,2
70 print"[CLR-HOME]liststop=aan."
```

Vic ondersteboven

Alle Vic-20 karakters op zijn kop afdrukken.

```
 1 rem ondersteboven / vic 20
 2 rem door jeroen rijsdijk
 3 rem uit spijkenisse
 4 rem
10 fora=0to63:forb=0to7:poke7168
   +a*8+b,peek(32768+a*8+(7-b)):
   next:next:poke36869,255
```

Vragen van Gebruikers.



Als je de statistieken bekijkt, die de vrijetijdsbesteding weergeven, gemeten over de seizoenen, dan zie je, dat er in de zomermaanden de meeste tijd wordt besteed aan surfen, kamperen, sporten en wat al niet meer. Computers staan statistisch in die tijd op een laag pitje. De werkelijkheid is echter anders, want de vragen stromen binnen. Of het komt door de slechte zomer, of door het feit, dat de Commodore meer aantrekkingskracht heeft dan in de grafieken staat, is mij niet bekend. De feiten (en de vragen) liggen er, laten we daarom maar snel van start gaan.

Rekenfouten van de C-64.

R.van Neutegem, Rozenburg probeerde onlangs z'n computer wat kwadraten te laten printen. Tot z'n grote schrik bleek er bij het berekenen iets fout te gaan, want bij 7 in 't kwadraat stond er als uitkomst geen 49, maar 49.0000001 en dat klopt niet. Hij betwijfelt nu de betrouwbaarheid van z'n hele Commodore.

Niet alleen deze C-64 rekent 'fout', maar alle 8 bits computers met digitale opslag maken deze gekke berekeningen. Je hoeft dan ook niet zo erg teleurgesteld te zijn in de mogelijkheden van de computer. De oorzaak hiervan is, dat de C-64 de numerieke variabelen (floating point), in z'n geheugen opslaat als waarden met een lengte van 5 bytes. En je begrijpt, dat je in vijf 8-bits posities lang niet alle getallen met decimalen zondermeer kunt onderbrengen. Daarom hebben de heren van Microsoft, die de Basic hebben geschreven, hier een truuk voor bedacht, waarbij gewerkt wordt met exponenten en mantisses, zodat in principe alle waarden in die vijf bytes passen. Het omrekenen van de reele waarde naar deze floating-point is erg gecompliceerd. Daar zal ik nu niet op ingaan. Bij deze rekenarij kan het niet anders of er moet wel eens iets worden afgerond. Bij het terugvertalen wordt deze afronding uiteraard niet gekorrigeerd en daardoor krijg je deze onverwachte getallen op je scherm. Bij het schrijven van software waarin korrekt moet worden gerekend, moet je dan ook terdege rekening houden met deze handicap. Je kunt dit doen door b.v. van alle getallen de INT-waarde te nemen.

Konstante 220 volt spanning?

R. da Thesta uit Den Helder vraagt of het nodig is, om tussen de C-64 en de netspanning een spanningsregelaar aan te sluiten. Dit zou zorgen voor 'schone' stroom, en de door fabrieken veroorzaakte spanningsruis uitfilteren.

Bij grote, commerciele, computersystemen, die in een omgeving worden gebruikt, waar de kans op spanningsfluktuaties en -storingen bestaat, gebruikt men meestal een spanningsstabilisator, om problemen met de voeding te voorkomen. Als het echter alleen een huis-configuratie betreft, is er relatief zeer weinig sprake van 'vuile' electriciteit. Daarom is het niet nodig om preventief een soortgelijk apparaat tussen net en computer te plaatsen. Pas als gebleken is, dat de locale stroomvoorziening veel te wensen overlaat, zou je de aanschaf van een stabilisator/filter kunnen overwegen. De prijs van zo'n apparaat ligt tussen de 200 en 500 gulden.

Return without Gosub.

Bert Jansen uit Delft heeft ons programma Bankafschriften ingetypt en daarna aangepast aan z'n EP 22 printer. Nu blijkt dit programma tot z'n ergenis een 'return without gosub error' te geven in plaats van te werken. Zijn vraag: hoe kan dat.

Als je alle regels uit het programma letterlijk hebt overgetypt, en met de checksum hebt gekontroleerd, kan deze fout beslist niet voorkomen. Ik denk dat de bug is ontstaan, doordat jij op de regels, die naar de subroutines verwijzen, hier en daar in plaats van 'GOSUB' de opdracht 'GOTO' hebt ingetypt. Dat is de enige oorzaak, die -bij een gestruktureerd programma- een soortgelijke foutmelding kan geven. Kijk de listing er nog maar eens op na.

Autorun.

In C-Info, nummer 5 stond op blz.73 een programma, waarmee je je eigen programmatuur automatisch kon laten starten.
H.B.Bijlsma uit Stadskanaal heeft de nodige problemen met de gebruiksaanwijzing van deze subroutine.

De wijze waarop je dit programma van Leon v.d. Torre moet gebruiken is als volgt: Eerst typ je de listing over en vervolgens run je het programma 'autorun'. Daardoor worden de nodige gegevens in machinetaal weggezet naar geheugenadressen waar de Basic-programma's niet kunnen komen. Daarna typ je je eigen Basic-programma's in (je kunt ze natuurlijk ook van tape of disk halen). Om nu de programma's met autorun te SAVEN, maak je geen gebruik van het SAVE-commando, maar je doet dit door te typen: SYS 36864 "programma-naam" RETURN. Op die manier wordt er automatisch het nodige met je Basic-programma gedaan, zodat het zichzelf zal starten als je het de volgende keer weer in wilt laden. Denk er wel om, dat je een kopie van je programma bewaard, zonder autorun, voor het geval je later nog eens iets aan de software wilt veranderen.

Serieel en Parallel

Wim Veldhuizen uit Groenlo staat voor een kostbaar probleem, omdat hij naast z'n C-64 beschikt over een parallelle diskdrive en een seriele printer. Als hij werkt met de SFD-1001 drive, die parallel wordt bestuurd door de IEEE-488 interface van Condor, gaat alles goed, tot het moment dat er gegevens moeten worden uitgeprint. Omdat dit serieel gebeurd, moeten deze gegevens eerst op een datasette worden gezet, dan moet de computer uit en zonder interface en drive weer worden opgestart.

Helaas Wim, een erg vervelende omstandigheid. De door jou gebruikte IEEE-488 is een erg verouderd interface, dat is ontwikkeld om de PET/CBM randapparaten te kunnen gebruiken. Er is daarbij nooit overwogen ook de seriele poort nog actief te houden. De enige mogelijkheid is de aanschaf van een beter interface, waarbij je er op moet letten dat er (softwarematig) kan worden omgeschakeld van parallel naar

serieel. Ik wil hier geen merken noemen, maar er zijn er diverse besproken in de vorige nummers van dit blad.

Print en print#.

John Giskes uit Spijkenisse stuurt ons een listing van het programma Adresbak, waar hij, ondanks grondige kontrole, geen fouten in kan ontdekken, en toch krijgt hij, bij het runnen een syntax-error als er gegevens naar de cassette worden geschreven.

John, het programma dat jij opstuurde, is als listing helemaal korrekt. De reden waarom het programma bij jou een fout geeft in regel 5860 ligt aan de manier, waarop je de listing hebt ingetypt. Zoals veel anderen, zul je in plaats van PRINT te toetsen het '?' gebruiken, want dat gaat veel sneller. De computer maakt echter wel degelijk onderscheid tussen PRINT en PRINT#, zoals dat in regel 5860 staat.
Deze beide commando's lijken uiterlijk veel op elkaar, maar voor de Basic-interpreter zijn het twee verschillende commando's. Daarom moet je ieder PRINT-statement met een hokje erachter voluit intoetsen, en geen vraagteken gebruiken. Wil je toch iets sneller typen, dan kun je voor PRINT# typen 'P shiftR', dat heeft hetzelfde effect.

Joysticktest.

Jan Velthuis is op zoek naar een methode om de joystickpoorten in Basic te kunnen uitlezen, zonder steeds weer een subroutine daarvoor te moeten aanroepen. Hij vraagt ons, of er niet een snellere methode is, om dit te doen, waarbij ook de diverse richtingen direkt kunnen worden bepaald.

Zoals bekend kan de inhoud van het geheugenadres dat korrespondeert met de joystickpoort via een PEEK worden uitgelezen. De adressen zijn 56320 voor poort 2 en 56321 voor poort 1. De waarde die dan gevonden wordt in dit adres, moet getest worden, om de beweging van de stick te kunnen bepalen. In Basic is dit meestal niet snel genoeg. Wil je werkelijk flitsend met je joystick aan het werk, dan ontkom je niet aan een machinetaal-routine. Sjoerd Bakker zal hier binnenkort het een en ander over uit de doeken doen. De beste oplossing in Basic komt van Menno Willemse uit Beverwijk. Hij heeft de poorttest in een paar functies ondergebracht.

```
DEF FNX(X) =((PEEK(56320) AND 4)=0)-((PEEK(56320)AND8)=0)
DEF FNY(X) =((PEEK(56320) AND 1)=0)-((PEEK(56320)AND2)=0)
DEF FNV(X) =((PEEK(56320)AND16)=0)
```

Door deze drie regels vooraan in je programma op te nemen, kun je vervolgens door b.v. A=FNX(0) in A bekijken of er naar rechts of links wordt bewogen (A=-1 of 1 of 0 resp links recht en stil). Hetzelfde voor A=FNY(0) voor hoog en laag en A=FNV(0) voor de vuurknop. De ruimte laat me niet toe hier dieper op in te gaan, maar deze methode staat wel sjiek in je progamma en werkt minstens zo snel als een subroutine met de nodige IF-tests. Menno bedankt en we zien de eerste programma's met deze implementatie met spanning tegemoet.

Killen van cartridges.

J.Pinkster uit Weesp wil graag de mogelijkheid hebben om z'n EPROMs in de expansiepoort te kunnen uitschakelen.

Het 'killen' van zelfstartende software uit cartridges is geen eenvoudige zaak. In feite kan het niet, als de software-maker er geen mogelijkheid toe heeft geboden. Er bestaan bepaalde toolkits, waarbij het mogelijk is door het ingeven van een bepaald commando, maar bij de meeste is dit een deel van de beveiliging, dus moet je toch je computer uitzetten, de cartridge eruit halen en vervolgens zonder uitbreiding verder gaan.

Saven na out of memory.

Een tip van J.P.Daniels uit Cuyck. Als je na het commando SAVE de foutmelding 'out of memory' op je scherm krijgt, kun je dat verhelpen, door POKE 56, 208. Daarna kan het bewuste programma, met naam en al worden weggeschreven.

Ik vraag me alleen af, wanneer zoiets nou nodig is.........!

Print using.

Dhr. Van Eijken uit Den Haag stuurt ons de printout van een factuur, die gemaakt is met een door hem geschreven programma. Op de factuur staan de bedragen niet in nette kolommen, wat een rommelig effect heeft. Zijn vraag: hoe is dit op te lossen.

In het allereerste nummer van C-Info stond een allround routine om dit te kunnen doen. Binnenkort zullen we dit programma opnieuw publiceren. Tot dan toe de volgende oplossing.

```
BE$ =STR$(INT(BE★100+.05)):LE-=LEN(BE$)
BE$=RIGHT$(" "+LEFT$(BE$,LE-2)+"."+RIGHT$(BE$,2),12)
```

In deze regels wordt BE$ de geformatteerde string van het bedrag BE. Het getal 12 geeft de totale breedte van de kolom aan. Dit kan uiteraard worden aangepast.

Compatibiliteit.

Dhr.J.C.Ooms te Bergen op Zoom heeft een CBM 8096, en wil graag weten, of het mogelijk is om de programma's van de C-64 om te schrijven, zodat ze werken op zijn business machine.

Helaas is het in de meeste gevallen niet mogelijk om (goede) C-64 programma's te vertalen naar de 8096. Dit komt, omdat de 64 over nogal wat extra's beschikt, zoals de VIC en SID chip, die niet zijn ingebouwd in de CBM business machines. Simpele Basic- programma's, zonder SYS, PEEK en POKE kunnen wel op alle machine draaien, maar daar heb je weinig aan, omdat deze software doorgaans al beschikbaar was voor de 80xx serie, toen er van een C-64 zelfs nog geen sprake was.

Schermdisplay op video.

Een brief van J.F.Wijntjes uit Rotterdam. Hij zoekt al enige tijd naar een mogelijkheid om de aktie op het scherm te kunnen opnemen op z'n videorecorder, zodat hij er ook van kan genieten, zonder zelfs maar een joystick aan te hoeven raken. Zijn probleem daarbij is, dat hij niet de juiste kabel kan vinden, die computer en recorder met elkaar verbindt.

Dit hoeft niet zo ingewikkeld te zijn. Je kunt de RF-uitgang (via een COAX-kabel) direkt op de video antenne ingang aansluiten, en je recorder afstemmen op kanaal 42-45. De kwaliteit die je dan krijgt is dezelfde als die van een normale TV-ontvanger, zonder monitor-ingang.
Als dit niet voldoende oplossend vermogen geeft, naar jouw zin, dan moet je de AUDIO/VIDEO uitgang gebruiken van zowel C-64 als recorder (ingang voor camera ed.) Deze manier is uiteraard beter, en feitelijk niet moeilijker dan de vorige. Je hebt daarvoor een dinplug nodig die past op de C-64. Verder een stuk afgeschermde 2-aderige (video)kabel en een aansluiting die past op de videorecorder. Deze dingen zijn standaard, dus in iedere radiohandel te koop. Je sluit op de C-64 pin 2 aan op de afscherming. Pin 4 (composiet

video) op de video ingang van de recorder en pin 3 gaat naar audio in. Standaard zijn deze kabels hier en daar wel te koop, met 3 tulppluggen aan een uiteinde. Deze moet je dan aan de video-recorder zien te koppelen.

Schermlokaties.

R.A.L. Terhoeve uit Oudetange wil graag meer informatie over de schermadressen van de C-16. Hij wil deze gebruiken om direkt karakters naar het scherm te kunnen poken.

De schermlokaties van de diverse Commodore-computers lopen nogal uiteen. En Commodore is niet erg gul met informatie over deze dingen. Vooral niet, waar het de C-16 en Plus/4 betreft.
Voor de volledigheid hier de schermlokaties van Commodore- machines.

VIC 20	$8000 - $8FFF	32768 - 36863
C-64	$0400 - $07E7	1024 - 2023
PET/40XX	$8000 - $83E7	32768 - 33767
CBM 80XX	$8000 - $87CF	32768 - 34767
CBM 8296	$8000 - $9FFF	32768 - 40959
C-16	$0C00 - $0FE7	3072 - 4071

Boekhouden

Ik heb voor mijn 64 een boekhoudpakket aangeschaft om er mijn eigen privé boekhouding en belastingaangifte mee te doen. Het programma werkt prima, maar ik heb nogal moeite om tussen alle ingewikkelde zakelijke boekhoudtoestanden de hoofdlijnen te herkennen en de zaak zo in te richten, dat het voor mij hanteerbaar wordt. Eigenlijk heb ik het gevoel, dat ik eerst een boekhoudcursus zou moeten volgen.

Wat betreft die cursus, geen gek idee, maar dat kost u wel flink wat avonden studeren. Helaas zijn vrijwel alle boekhoudpakketten voor professioneel gebruik afgestemd op de zakelijke gebruiker, compleet met facturen, overlopende rekeningen, ingewikkelde rekeningstelsels en een hele hoop ballast. Niet voor de zakenman, die heeft dat allemaal nodig, maar wel voor de particulier, die in wezen alleen bij wil houden wat er binnenkomt, uitgaat en wat er op zijn aangifteformulier moet komen. Er zijn natuurlijk ook wel huisboekhoudpakketjes, maar die komen weer te kort en zijn bovendien meestal niet afgestemd op onze lokale situatie. Toch ligt daarin misschien wel een oplossing. Schaf eerst eens zo'n pakketje aan en experimenteer daar wat mee, dan wordt de structuur van een boekhouding toch wat duidelijker. Daarna kunt u overstappen op een "echt" pakket.

De Nederlandstalige boekhoudpakketten zoals Finad64 en Boekhouden van Prompt zijn uitstekend voor het zwaardere werk, maar u mist daarin de grondslagen van het boekhouden. Men gaat uit van enigszins ervaren boekhouders. Ook de voorbeelden zijn al gauw te ingewikkeld. Maar in de praktijk zal blijken, dat wanneer u door de brij aan het begin heen bent, het zich verder redelijk vanzelf wijst. Die beginfase houdt oa. het opstellen van het rekeningstelsel in. Daarbij gaat het om een paar grote categorieën, die van belang zijn. Verlies en Winst rekeningen voor wat u binnenkrijgt (verdient) en wat u uitgeeft, dus de kosten van auto, porti, briefpapier etc. Dan de rekeningen voor uw bezit (en schulden), dus giro, bank, uw computer (en de software), uw auto en andere apparatuur. Als laatste het kapitaal, maar dat rekent de computer wel voor u uit. Probeer gewoon eens een eenvoudige opzet en probeer het simpel te houden. De vragen naar BTW en facturen etc. zijn voor iemand met een kasboekhouding zonder BTW afdracht niet van belang, kies dan bv. altijd het 0% tarief. Het is niet echt nodig om u in het begin bezig te houden met afschrijvingen, WIR premies etc. Dat kunt u aan het eind van het jaar regelen of laten regelen door uw boekhouder/accountant. Want die zult u meestal toch wel nodig blijven houden, als een soort zeef aan het eind. Maar raadpleeg hem ook vooraf eens, vooral bij het opzetten van het rekeningschema.

Uw probleem is niet helemaal opgelost en er blijft dus behoefte aan een goede computergestuurde cursus boekhouden! Wie maakt er eens een goede Tutor bij een bestaand programma?

Disk-capaciteit.

In Uithuizen zit Guide Bruin zich af te vragen, of je niet beter dubbel zijdige 96 TPI diskettes kunt gebruiken bij je 1541, waardoor dan de opslagmogelijkheid op een diskette meer dan verdubbeld wordt, omdat er veel meer tracks op zo'n diskette kunnen worden geschreven.

Guido, ik weet niet waar je deze wijsheid vandaan haalt, maar het is helaas niet zo, dat je door een betere kwaliteit diskette te gaan gebruiken automatisch ook meer op zo'n schijf kwijt kunt.
De 1541 werkt op zowel enkelzijdige als dubbelzijdige disks. Het verschil zit hem in de kwaliteit van de schijf, zodat je waarschijnlijk minder problemen tegen zult komen, als je betere disks gebruikt. De 1541 heeft een schrijfdichtheid, die hoger ligt dan 'single density' (50 procent meer). Het is echter niet de disk maar de drivekop die bepaalt hoeveel data er (geformatteerd) naar een diskette kan worden weggeschreven. Wil je dus een grotere diskopslag, dan zul je ook een andere drive nodig hebben.

Tot zover de problemen van deze keer. Ik hoop dat je er wat wijzer van bent geworden. De achterstand in het beantwoorden zijn we min of meer te boven, maar dat komt voor een groot gedeelte, omdat er de laatste tijd nogal wat artikelen in C-info zijn verschenen, die al of niet direkt betrekking hadden op vragen van de lezers. Heb je dus geen antwoord op je vraag gekregen, dan kun je er bijna zeker van zijn dat ergens in C-info wel iets over je probleem te lezen staat. Verder heb ik nogal wat telefoontjes gepleegd om specifieke vragen te helpen oplossen.
Wil je iets van ons weten of heb je een probleem, omschrijf je vraag dan liefst zo duidelijk mogelijk, en stuur het op aan de redaktie, t.a.v. Jan Bodzinga. Vergeet niet te vermelden dat het deze rubriek betreft, en als het kan zet dan ook je telefoonnummer erbij. Dat alles garandeert een snelle oplossing van je problemen.

Jan Bodzinga.

Het spelen met Karakters ofwel het maken van een eigen tekenset kan vele uitingen krijgen. Van Quasarsoft, waar de heer S.Pool achter steekt, kregen we een heel uitgebreide versie met veel uitleg.

C 64

Lett Lett Lettervergroting

Zelf experimenteren met grotere letters, dat wil vrijwel iedereen die eigen software maakt. In dit artikel komt dat aan de orde aan de hand van een programmalisting.

Het Programma

Het programma vraagt allereerst om het formaat van de uitvoer. Hier moet x,y ingevoerd worden, waarin x de breedte voorstelt, en y de lengte. Dan moet de te vergroten string ingevoerd worden. Tenzij men zelf aan het knutselen gaat, moet men zien te voorkomen iets anders dan de normale letters en spaties te gebruiken (dus geen graphics/leestekens), omdat slechts de ASCII tekens worden gebruikt en niet de speciale CBM tekenset.

Printen

Vervolgens wordt om invoer van het output-device gevraagd. Dit spreekt voor zich. Voor de printer wordt device #4 gebruikt, dit kan natuurlijk veranderd worden, al naar gelang de gebruikte printer. Het programma is geschreven voor een Seikosha GP-100 VC, en moet dus zondermeer te gebruiken zijn voor de MPS-801 etc. Met een beetje experimenteren zal het wel lukken. Echter wel letten op regel 1250 want daar staan enkele printer-specifieke codes (zie uitleg). Het gaat er met name om te voorkomen, dat de printer teveel line-feeds geeft.

Als laatste wordt de wijze van uitvoer gevraagd. Dit kan zowel horizontaal als vertikaal. Horizontaal wil zeggen dat het papier/scherm op z'n kant gehouden moet worden om het te lezen. Bij vertikaal verschijnen de letters leesbaar onder elkaar. Na invoer van het laatste gegeven gaat de computer aan het werk, waarbij de snelheid bijna alleen bepaald wordt door de printer, en het formaat van de letters. Bij gebruik van het scherm is enige vertraging als gevolg van grote letters bijna niet te merken.

Details

Zonder op al te veel details in te gaan, moet de in de techniek van het programma geinteresseerde lezer toch het volgende weten om het programma te snappen.
Als de C-64 normaal werkt, wordt het scherm opgebouwd door VIC II (Video Interface Chip 6567). Dit is de chip die zorgt voor alle soorten beelden die op het scherm verschijnen. Deze zijn: Standard-, Multicolor-, en Extended Background Color karakters; High-Res; Multi- en Singlecolor Sprites; Screen-Blanking (bij tape I/O). Verder heeft deze chip nog meer functies, zoals een Dynamic-Ram-Refresh; Light-Pen ondersteuning; Interrupt-handling en control en Smooth- Scrolling. We zullen ons nu beperken tot de Standaard-Karakters.

Zoals bekend ligt normaal de Video-Matrix van 1024-2023. Daar staan in de geheugenplaatsen de Screen-Codes van de karakters die op dat moment op het scherm staan. Deze Screen-Codes varieren van 0-255. Echter, wie het beeld op de monitor goed bekijkt, ziet dat deze zijn opgebouwd uit kleine puntjes. Bij de CBM-64 bestaan alle karakters uit een matrix van 8 bij 8 puntjes (pizels). Dat zijn 64 puntjes, en hoe bepaal je nu met 1 Byte (=8 bits) welke van die 64 puntjes aan staan?

Screen Code

Wel, dit is opgelost door alle karakters een code te geven (de Screen-Code), en van ieder karakter de gegevens over de opbouw hiervan ergens anders in een soort tabel te zetten. Iedere keer als de VIC II nu een bepaald karakter moet laten zien (volgens de Video-Matrix) zoekt hij in de tabel op hoe die is opgebouwd. Deze tabel heet Character-Memory, en is normaal geplaatst van D000-DFFF (hex.). Hoe is dat geheugen nou samen gesteld? Bijvoorbeeld de letter A (van 53256-53263=D008-D00F):

SCHERM	BINAIR	DECIMALE
**	00011000	24
****	00111100	60
** **	01100110	102
******	01111110	126
** **	01100110	102
** **	01100110	102
** **	01100110	102
	00000000	0

Zoals hieruit valt af te leiden, stelt iedere Byte 1 rij puntjes van het karakter voor, en is iedere bit die 1 is representatief voor één oplichtend puntje op het scherm.

Voor ieder karakter zijn 8 Bytes nodig en de tabel is gerangschikt naar Screen-Code, dus de plaats van deze bytes is simpel te berekenen:
Begin + 8 ⋆ Screen-Code.
Hier is Begin de start van het Character-Memory, welke normaal op D000 staat, maar wel veranderd kan worden. Op dit veranderen zal ik nu niet ingaan, maar dit verplaatsen van het Character-Memory geeft je de mogelijkheid om zelf je eigen karakter-sets te maken.

Het programma berekent dus eerst het start-address van het te printen karakter, en test dan bit voor bit of deze aan staan. Mocht het niet duidelijk zijn hoe dit gedaan wordt, dan moet men de handelingen van regel 450-530 maar eens uitschrijven. Dan wordt nl. het effect van de instruktie AND duidelijker.

Subroutines

De subroutines waar een zeer belangrijke handeling wordt verricht zijn de I/O-IRQ OFF/ON routines. Hier wordt namelijk het Input/Output geheugen gebied van het Character-Memory (in ROM) getrokken. Bovendien wordt zo de hardware interrupt uitgeschakeld, waardoor er niet in de Char-ROM gekeken wordt door de diverse systeem routines, en wij ongestoord kunnen PEEKen. Anders zouden er heel rare dingen kunnen gebeuren.

Nadeel van deze POKE's is dat we geen I/O meer hebben (dus geen printer of iets dergelijks kunnen aansturen), en ook geen invoer meer kunnen plegen door middel van het keyboard. Daarom moet er continu worden geswitched, om uit-invoer mogelijk te houden.

Waarschuwing: ga niet zomaar experimenteren met deze POKE's als er een nog niet geSAVEd programma in het geheugen zit.

Is nl. de I/O-IRQ uitgeschakeld, dan kan deze alleen binnen een programma, of door uitschakelen worden hersteld. Daarom: wees zeer voorzichtig met het zomaar onderbreken van het programma als het loopt, want de kans bestaat dat er wel READY/BREAK komt, maar ZONDER knipperende cursor. Hierdoor kan er niets ingevoerd worden, en werkt STOP/RESTORE ook niet (in dit geval alleen als ook de I/O uitgeschakeld is).

Zoals eerder gezegd, kan dit programma alleen letters en spaties afdrukken. Dit komt, omdat er maar een beperkt verband is tussen de Screen-Codes, en de ASCII-Codes, welke gebruikt worden om letters op te slaan, en te printen. De BASIC functies ASC("X"), en chr$(X) hebben betrekking op de ASCII-codes, terwijl voor een POKE in het schermgeheugen de Screencode nodig is.

Het probleem zal duidelijk worden na een blik in de Users-Manual, in de appendices E en F. Bij de ASCII(F) beginnen de letters bij #65, terwijl dat bij de Screen-Codes bij #1 is.

Voor de letters geldt: Screen-Code=ASCII-64. Doch zodra er cijfers worden ingevoerd, of speciale tekens komt er garbage op het scherm, of er verschijnt een karakter dat niet gewenst is. In het programma wordt namelijk gewerkt met de functie: ASC(X$), en zoals vermeld, levert deze functie de ASCII-codes.

In het programma is als geringe opheffing van het ongemak, voor de spatie wel een voorziening getroffen, en dit zal toereikend blijken voor het maken van simpele teksten. Doch wil men volledig gebruik maken van de enorme karakter-set van de C-64 (512 tekens) dan moet een andere methode gebruikt worden.

Alternatief

Als alternatief zal een voorziening getroffen moeten worden om het programma van de Screen-Codes te voorzien. In BASIC echter, is er hiervoor geen standaard functie. Oplossing is natuurlijk door middel van PEEKs in het schermgeheugen de Scr.Cd. te weten te komen.

Probleem: hoe weet ik waar precies de letters staan?

Dit kan op meerdere manieren worden opgelost, nl. door de string op een bekende plaats te zetten, bv. boven aan het

DE LISTING:

```
10 PRINT"♥"
30 PRINTTAB(10)"*********************"
50 PRINTTAB(10)"*  BIG CHARACTERS  *"
70 PRINTTAB(10)"*     GENERATOR    *"
90 PRINTTAB(10)"*   (C) BY S.POOL  *"
110 PRINTTAB(10)"*  '85 QUASAR-SOFT *"
130 PRINTTAB(10)"*********************"
150 GOSUB1490:I1$="▒▒▒▒▒▒▒":I2$="       "
170 I3$="":FORA=1TO7:I3$=I3$+I2$:NEXT
190 INPUT"QQ WHAT SIZE MUST THE CHAR.BE(X,Y)";X,Y:IF(X>5)OR(Y>5)THEN190
210 INPUT"QQ WHICH STRING MUST BE DISPLAYED";SR$
230 INPUT"QQ OUTPUT TO RS■CREEN OR RP■RINTER";O$:IF(O$<>"S")AND(O$<>"P")THEN230
250 IFO$="P"THENOPEN1,4
270 INPUT"QQ RH■ORIZONTAL OR RV■ERTIKAL OUTPUT";K$
290 IFK$="V"THEN430
310 IFK$="H"THEN550
330 GOTO270
350 :
370 REM *******************
390 REM * START VERT.LOOP *
410 REM *******************
430 GOSUB1390:FORC=1TOLEN(SR$)
450 DA=53248+8*(ASC(MID$(SR$,C,1))-64):IFMID$(SR$,C,1)=" "THEN810
470 FORW=DATODA+7:OU$="":R=PEEK(W):FORS=7TO0STEP-1
490 IF(R AND 2↑S)THENOU$=OU$+LEFT$(I1$,X):NEXTS:GOSUB1050:NEXTW:GOTO730
510 OU$=OU$+LEFT$(I2$,X):NEXTS:GOSUB1050:NEXTW
530 NEXTC:CLOSE1:GOSUB910:RUN
550 :
570 REM ***********************
590 REM *** START HOR.LOOP ***
610 REM ***********************
630 GOSUB1390:FORC=1TOLEN(SR$):DA=53248+8*(ASC(MID$(SR$,C,C))-64)
650 IFMID$(SR$,C,C)=" "THEN810
670 FORW=7TO0STEP-1:OU$="":FORS=7TO0STEP-1:R=PEEK(DA+S)
690 IF(R AND 2↑W)THEN OU$=OU$+LEFT$(I1$,X):NEXTS:GOSUB105A:NFXTW:GOTO730
710 OU$=OU$+LEFT$(I2$,X):NEXTS:GOSUB1050:NEXTW
730 NEXTC:CLOSE1:GOSUB910:RUN
750 REM ************************
770 REM *** SPACE SUBROUTINE ***
790 REM ************************
810 IFO$="S"THEN850
830 FORA=1TO8*Y:PRINT# 1:NEXTA,C:GOSUB910:RUN
850 FORA=1TO8*Y:PRINT:NEXTA,C:GOSUB910:RUN
870 :
890 :
910 REM ********************
930 REM *** WAIT ROUTINE ***
950 REM ********************
970 GOSUB1490:PRINT"QQ   PRESS ANY KEY PLEASE"
990 GETA$:IFA$=""THEN990
1010 GOSUB1450:RETURN
1030 :
1050 REM **********************
1070 REM *** CHOOSE OUTPUT ***
1090 REM **********************
1110 IFO$="P"THENGOSUB1170:RETURN
1130 GOSUB1290:RETURN
1150 :
1170 REM **********************
1190 REM *** PRINTER OUTPUT ***
1210 REM **********************
1230 GOSUB1490:FORA=1TOY:IFOU$=LEFT$(I3$,8*X)THENPRINT#1:NEXT:GOSUB1390:RETURN
1250 PRINT#1,CHR$(15)OU$CHR$(8):NEXT:GOSUB1390:RETURN
1270 :
1290 REM *********************
1310 REM *** SCREEN OUTPUT ***
1330 REM *********************
1350 FORA=1TOY:PRINTOU$:NEXT:RETURN
1370 :
1390 REM ******************
1410 REM *** I/O-IRQ OFF ***
1430 REM ******************
1450 POKE56334,PEEK(56334)AND254:POKE1,PEEK(1)AND251:RETURN
1470 REM *****************
1490 REM *** I/O-IRQ ON ***
1510 REM *****************
1530 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1:RETURN
1550 REM END

READY.
```

scherm, door eerst HOME, en dan de string te printen, of, en dit is eleganter, door de positie van de INPUT instruktie te bepalen. Want iedere keer als er wordt gevraagd om een string, staat het vraagteken, en dus de in te voeren string op de zelfde plaats. Is deze plaats bekend, dan kunnen eenvoudig deze plaatsen gePEEKt worden.

Iets waar ook rekening mee moet worden gehouden, is het feit dat het scherm Scrollt als de letters naar het scherm worden uitgevoerd. Hierdoor verplaatst de string zich naar boven, en zijn de PEEK-adressen niet meer van toepassing. (Hint: gebruik een ARRAY om de Screen-Codes op te slaan, voor dat met de werkelijke uitvoer begonnen wordt).

Het programma is geheel modulair opgebouwd, en is dus als zodanig makkelijk te volgen. De printer specifieke codes waarover werd gesproken, zijn: CHR$(15); en CHR$(8). Deze codes dienen om de linefeed na het printen zo te regelen dat de uitvoer één geheel vormt.
In het geval dat de gebruikte printer niet CBM-compatibel is (of andere codes worden gebruikt) moet er in de manual gezocht worden naar vergelijkbare codes. Experimenteren kan geen kwaad! (De linefeed moet dus na het printen van een balkje beperkt blijven tot 1 karakter hoogte.)

Een laatste tip, deze techniek kan ook worden gebruikt voor het afdrukken van Sprites en eigen tekensets.

MISSERS

We komen er niet onderuit. Of het nu aan het slechte vakantieweer ligt, of aan de drukker, de zetter of de computer, er blijven hier en daar wat fouten zitten in de listings uit ons blad. Maar goed dat we beschikken over "Missers", zodat we de zaak snel recht kunnen zetten. Overigens onze welgemeende excuses, en we doen er alles aan, om zo weinig mogelijk mis te schieten.

Natuurwonder.

In Jaargang 2 nr.7 stond een listing van het programma 'Natuurwonder'. Er zitten nogal wat onnauwkeurigheden in, die niet allemaal grote bugs tot gevolg hebben, maar ook een paar, waardoor het geheel niet loopt.
35 PRINT "P = DRUK": PRINT "A = OPPERVLAKTE"
De regels 60 tot 100 moeten worden vervangen door:
60 IF A$="m"THEN 110
65 IF A$="d"THEN 190
70 IF A$="v"THEN 270
75 IF A$="g"THEN 350
80 IF A$="z"THEN 430
85 IF A$="s"THEN 750
90 IF A$="k"THEN 510
95 IF A$="a"THEN 670
100 IF A$="p"THEN 580
105 RUN
regel 160 wordt 160 PRINT"De massa is";c;" kg"
In regel 240 moet er in plaats van C$ een C als variabele staan.
Regel 535 invoegen: 535 INPUT"Oppervlakte in m2";b
Verder kunnen er nog wat spaties in de printregels worden aangebracht, om het geheel op het scherm een beter aanzien te geven. Voor de volledigheid, 'Newton' mag worden afgekort tot de hoofdletter 'N'. Alle oplettende natuurkundigen, die ons hierop attendeerden, hartelijk dank, en nogmaals onze excuses.

Diskette ID

In de vragenrubriek van jaargang 2 nr.7 kwam het herstellen van de diskette-ID aan de orde. Daarbij is een programma afgedrukt, waarbij in regel 60 het commando print#15,"m-r:" wordt gebruikt. Er blijken ROM versies van de 1541 te bestaan, die een andere interpretatie van deze opdracht vereisen. De oplossing hiervoor is, de ':' achter m-r weglaten. Dan wordt het juiste gedeelte uit de drive-RAM nulpagina ingelzen en verwerkt. Beide mogelijkheden proberen, geeft uiteindelijk een goed resultaat.

Schermwerk

Over het artikel schermwerk kregen we helaas nogal wat boze brieven. De meeste ervan waren niet helemaal terecht, want de programma's die erbij waren afgedrukt zijn op een na korrekt. Je moet bij het runnen van deze software er alleen rekening mee houden, dat het ontzettend lang duurt, voordat alle bytes in RAM zijn opgesplitst en bewerkt. Het gaat hier uiteindelijk om een komplete karakterset. Dus na RUN zo'n 10 tot 13 minuten geduld, en je krijgt prachtige letters op je scherm.

In het programma 'BREDE LETTERS' van Ben van Volen zaten helaas wel een paar fouten:
regel 9 e.v. wordt:
9 D(Q)=D(Q)+VAL-(MID$(D$(Q),C,1))★2 ↑ X
10 NEXT X: C=0: D$(Q)="": NEXT Q: RETURN
14 POKE A+W,D(1):POKE A+U,D(2): D(1)=):D(2)=0:NEXT
Het programma is, na het veranderen van deze drie regels, helemaal foutloos en het resultaat mag er zijn.

J.B.

CES 1986

Nederlandse deelname aan Amerikaanse beurs voor consumenten electronica

De Nederlands-Amerikaaanse kamer van koophandel, in samenwerking met de EVD bereidt een gezamenlijke deelname van Nederlandse bedrijven aan de CES (9-12 januari Las Vegas USA) voor. Dat is een ideale springplank om bv. computersoftware of randapparatuur onder de aandacht van de Amerikaanse pers en inkopers te brengen.
Er is de mogelijkheid van subsidie en belangstellenden kunnen contact opnemen met de heer van Dijk. ☎ 070-478234.

Boekenrubriek

Luitingh computersoftware.

De (computer)boekhandel is sinds kort verrijkt met een drietal boeken voor de C-64, die worden uitgegeven door uitgeverij Luitingh te Utrecht. De werkjes bevatten stuk voor stuk programma's voor de 64 en kosten elk Fl.14.90

Als titel kregen ze respectievelijk mee:

Spelletjes voor je Commodore deel 1. (ISBN 90-245-1810-5)

Spelletjs voor je Commodore deel 2. (ISBN 90-245-1830-X)

Fantastische avonturen met je C-64. (ISBN 90-245-1820-2)

Ze zijn geschreven door diverse auteurs, waaronder Robert Young, die al een serie dergelijke (engelse) boeken op z'n naam heeft staan. De inhoud van de drie boekjes ligt in hetzelfde vlak, zodat ze onder dezelde noemer vallen.

Naast de listings hebben de boekjes een (identieke) woordenlijst van computertermen opgenomen en een stuk tekst waarin wordt uitgelegd hoe je je eigen programmeertechniek kunt verbeteren.

De boekjes zijn niet duur, en misschien wel gemakkelijk voor de beginnende computeraar. De listings zijn echter vrij rommelig gedrukt, iedere vorm van syntax-controle ontbreekt en ook de grafische en funktietoetsen zijn onduidelijk weergegeven.

Al met al niet het beste wat er ooit over Basic-programma's is verschenen, maar dat is geen wonder. De opdracht die de schrijver aan deel 1 heeft meegegeven spreekt voor zichzelf: 'Dit boek draag ik op aan mijn ego en mijn verlangen om snel rijk te worden.

The Repair and Maintenance Guide

door H.Beechhold

Dit Engelstalige werkje van 262 bladzijden zou de oplossing voor alle problemen moeten inhouden, en wie niets van elektronica weet, heeft er ook zeker iets aan. Maar zoals vaak met computerboeken, hoe algemener ze zijn, des te minder je er aan hebt. Want hoofdstukken lang doorzeuren over soldeertechnieken, schoonmaken, erg veel uitleg en definities over zaken die niets met onderhoud te maken hebben, dan zou je uiteindelijk toch wat harde informatie verwachten. Bijstellen van de cassettekoppen, schoonmaken van de toetsen, het wordt wel behandeld, maar erg oppervlakkig. Dit is een goed verzorgd en makkelijk te lezen (in het Engels) boek, en een goede introductie tot computerkennis voor de iets gevorderde. Als onderhoudsgids echter tamelijk beperkt, niet ingaan op de specifieke machines maakt het te vrijblijvend.

Machinetaal Routines voor de C-64

Uitgave Holt Saunders ƒ 39,50

Dit boek geeft een groot aantal machinetaal routines, die in de eigen Basic-programma's kunnen worden ingevoegd om wat meer snelheid in zijn programma's te brengen. Er is een hulpprogramma om het foutloos intypen te ondersteunen, iets wat eigenlijk in alle listingboeken verplicht zou moeten zijn, maar dat we nog te vaak missen.

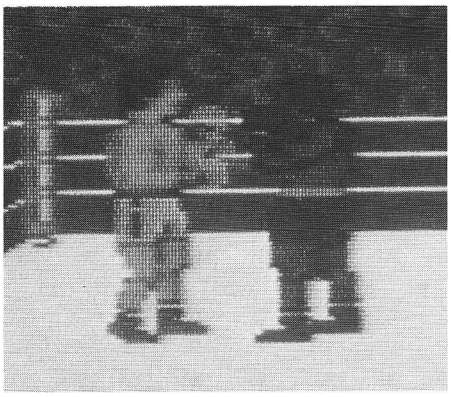
J.B.

Activision

of de computer te verslaan. U speelt over negen innings (slagbeurten) en bij gelijkspel worden dit er eventueel meer. Mocht u de computer verslaan dan mag u zichzelf een **Star Leaguer** noemen. Dat zal u echter niet licht vallen, daar de 64 ongelooflijk sterk speelt (af en toe bekruipt mij zelfs het gevoel dat hij vals speelt).
Al met al is dit werkelijk een fantastisch spel. Niet slechts vanwege de mooie graphics en het fraaie geluid maar voornamelijk om het spelplezier. En dat is wat er nu net aan een hoop spelen mankeert, het ziet er prachtig uit, klinkt mooi maar het verveelt al na een half uur. Met dit spel hoeft u daar dus niet bang voor te zijn. Een ware aanwinst voor de softwaremarkt.

Boksen

Vechtsporten zijn erg populair, maar naast de Oosterse varianten is er natuurlijk nog "The noble art of selfdefence", het aloude boksen.
Activision maakte een erg Amerikaans aandoend spel, met de naam van de wereldkampioen Lichtgewicht Barry McGuigan in de titelrol. Het is ontwikkeld door Troy Lyncon en men wil vooral de kunst van het boksen via de computer laten zien. Dat wil zeggen complexe bewegingsroutines, een arsenaal aan stoten, verdedigingen, voetbewegingen, stijlfiguren en strategie, alles in drie dimensies.



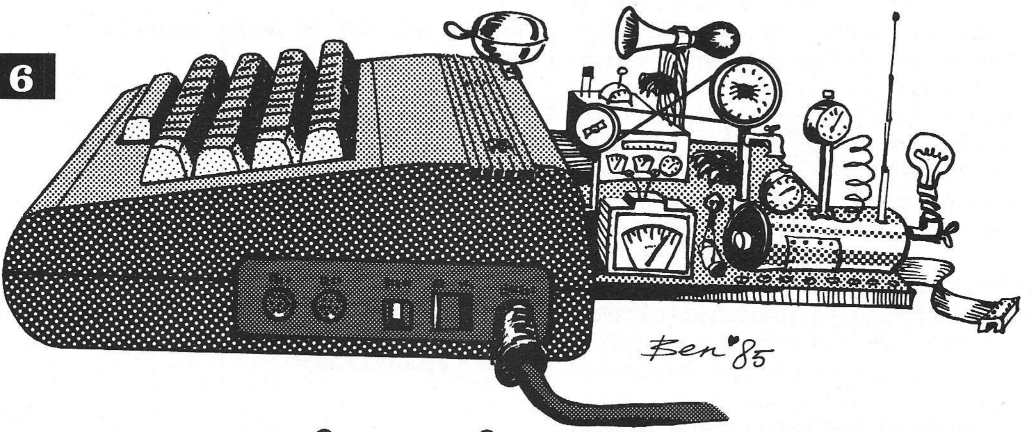
Men kan kiezen uit een heel peloton tegenstanders, ieder met een eigen karakter, maar ook zelf een kunstmatige bokser ontwerpen met een eigen stijl, huidskleur, grootte en gezicht, hetgeen een leuke en creatieve aanvulling is op het boksen zelf. De eigen bokser kan dan ook nog door training beter worden, zijn uithoudingsvermogen verbeteren en aan kracht winnen. En dan de ring in, met natuurlijk een juichend publiek, camera's en een achtergrondmuziekje. Een spel van grote klasse, grafisch erg verzorgd en van Amerikaanse Activision kwaliteit en die vinden we iets beter dan wat soms onder dat label in Europa uitkomt.

Summer Games II

Dit is een vervolg op de Summer Games tophit en zoals u begrijpt volgt hierna ook nog Winter Games. Van onberispelijke kwaliteit, men heeft aardig verdiend bij Epyx en een beetje zorg besteden aan hun produkten kan er dus wel vanaf. Deze keer weer andere sporten, zoals fietsen, schermen, roeien, hoogspringen, speerwerpen, paardrijden. Weer met de bekende achtergronden, ceremonies en gejuich. Dit programma heeft geen aanbeveling nodig en is de fl 49,50 dan ook wel waard. De kinderen beleven er veel plezier aan, dit is een combinatie van veel spelletjes in een pakket.

Machinetaal blijft een populair onderwerp, merken we aan uw reakties. In deze aflevering behandelt Sjoerd Bakker de synthesiser van de Commodore 64: de SID-chip. Met het programma van deze keer blijkt SID ook een meer dan 350 jaar oude canon ("Viva la musica" van Praetorius) verrassend nieuw te laten klinken.

machinetaal deel 6



Viva la musica! De SID-chip.

Toen de Commodore 64 in 1982 op de markt verscheen, was een van de meest opmerkelijke "features" de aanwezigheid van een 3-stemmige synthesiser: de SID-chip (6581 Sound Interface Device). Indertijd was zo'n muziekdoos voor homecomputers een unicum; tegenwoordig is het een standaardvoorziening bij de meeste merken. SID blijft echter gunstig afsteken bij de concurrentie. Naast muziek komt tegenwoordig ook al wat (gesynthetiseerde) spraak uit de 64, een welkome aanvulling bij vele spelletjes.

Het programmeren van de SID- en de VIC-chip is iets wat je, ook vanuit Basic-standpunt bekeken, voor grotere programma's beter kunt doen in machinetaal dan in Basic. In tegenstelling tot MSX-Basic kent "64-Basic" geen opdrachten als SOUND of SPRITE, waarmee je deze chips een afgerond stuk werk kunt laten doen. Je moet je dus behelpen met PEEK en POKE, en dat gaat in machinetaal veel sneller en efficienter.

Muziektheorie

Als je SID behalve geluidseffekten ook muziek wilt laten maken, is een zekere kennis van de muziektheorie gewenst. Kun je bijvoorbeeld noten lezen (wat echt niet zo moeilijk is), dan kun je ook elk 1-, 2-, en 3-stemmig stuk door de 64 laten spelen. Het geven van muziekonderricht past echter niet in het kader van een computerblad. De registers (geheugenplaatsen waarvan inhoudsverandering een automatisch gevolg met zich meebrengt) van SID beslaan het adresgebied $D400-$D41C. Decimaal is dat gelijk aan 54272-54300. De werking van deze registers is aanzienlijk eenvoudiger dan die van de CIA-chips uit de vorige aflevering.

Zoals gezegd kun je met de SID-chip 3 aparte stemmen "tegelijk" programmeren.

Deze stemmen leggen ieder beslag op een gedeelte van het registergebied, en verder zijn er nog 8 registers die betrekking hebben op de 3 stemmen gezamenlijk. Stem 1 loopt van $D400-$D406, Stem 2 van D407-D40D, Stem 3 van $D40E-$D414, en de gezamenlijke registers van $D415-$D41C. Omdat de werking van de drie groepen stemregisters voor elke stem hetzelfde is, zal ik als voorbeeld alleen de registers van stem 1 behandelen.

(Illustratie 1)

SID REGISTEROVERZICHT												
	Basisadres $D400 (54272) +				Bits							
Nr.	Stem1	Stem2	Stem3	Naam	7	6	5	4	3	2	1	0
$00	$00	$07	$0E	FREQ LO	F 7	F 6	F 5	F 4	F 3	F 2	F 1	F 0
$01	$01	$08	$0F	FREQ HI	F 15	F 14	F 13	F 12	F 11	F 10	F 9	F 8
$02	$02	$09	$10	PW LO	PW 7	PW 6	PW 5	PW 4	PW 3	PW 2	PW 1	PW 0
$03	$03	$0A	$11	PW HI	----	----	----	----	PW11	PW10	PW 9	PW 8
$04	$04	$0B	$12	CONTROL	NOI	PUL	SAW	TRI	TEST	R MO	SYNC	GATE
$05	$05	$0C	$13	ATK/DCY	ATK3	ATK2	ATK1	ATK0	DCY3	DCY2	DCY1	DCY0
$06	$06	$0D	$14	STN/RLS	STN3	STN2	STN1	STN0	RLS3	RLS2	RLS1	RLS0
$15				FC LO	----	----	----	----	----	FC 2	FC 1	FC 0
$16	Gezamelijke registers			FC HI	FC10	FC 9	FC 8	FC 7	FC 6	FC 5	FC 4	FC 3
$17				RES/FLT	RES3	RES2	RES1	RES0	FTEX	FLT3	FLT2	FLT1
$18				MOD/VOL	3OFF	HP	BP	LP	VOL3	VOL2	VOL1	VOL0
$19				POT X	PX 7	PX 6	PX 5	PX 4	PX 3	PX 2	PX 1	PX 0
$1A	Read-only			POT Y	PY 7	PY 6	PY 5	PY 4	PY 3	PY 2	PY 1	PY 0
$1B				OSC 3	O 7	O 6	O 5	O 4	O 3	O 2	O 1	O 0
$1C				ENV 3	E 7	E 6	E 5	E 4	E 3	E 2	E 1	E 0

FREQUENTIES

Register $00. FREQ LO

In dit register wordt de Lo-byte van de frequentiewaarde geladen. De frequentie van een toon bepaalt hoe hoog die toon zal klinken. Om redenen waar ik nu niet verder op in zal gaan, is het frequentiegetal dat je in de SID-chip stopt niet gelijk aan de frequentie zoals die uit je luidspreker komt. Ook komen de waardes van de frquentietabellen die Commodore in de User Manual en Reference Guide geeft niet overeen met de A=440 Hz-standaardwaarde. Je kunt deze tabellen wel gebruiken, maar als er gelijktijdig een ander instrument speelt, zal dit vals klinken.

Juiste frequenties

Bijgaand programma produceert een tabel met de juiste waardes voor 8 octaven (0-7). Eerst krijg je de naam van de betreffende noot met het octaafnummer, vervolgens de decimale waarde voor beide frequentieregisters gezamenlijk, daarna de waardes per register (Hi-byte, Lo-byte), en tenslotte de Hexadecimale waarde. De waarde van de hoogste noot (B 7) is onjuist, maar deze is dan ook te groot om in twee registers te passen. Met RUN gaat de output naar je printer; met RUN 110 naar het scherm.

```
100 OPEN 4,4: CMD4
110 READ A,B,K$,L$,N$,H$
120 PRINT K$ L$: PRINT: T=3
130 FOR OC=0 TO 7
140 FOR Q=1 TO 12: R=2*Q-1
150 PRINT SPC(2) MID$(N$,R,2) OC SPC(2)
160 X=2↑(T/12)*A/B: Y=INT(X+0.5)
170 HB=INT(Y/256): LB=Y-HB*256
180 W(1)=Y: W(2)=HB: W(3)=LB
190 FOR M=1 TO 3: S=0
200 IF W(M)<10000 THEN S=S+1
210 IF W(M)<1000 THEN S=S+1
220 IF W(M)<100 THEN S=S+1
230 IF W(M)<10 THEN S=S+1
240 W(M)=S: NEXT
250 PRINT Y SPC(3+W(1))
260 PRINT HB SPC(W(2)-1) LB SPC(1+W(3))
270 Z=Y: D=4096: PRINT "$";
280 FOR C=1 TO 4: P=INT(Z/D)
290 Z=Z-P*D: T$=MID$(H$,P+1,1)
300 PRINT T$;: D=D/16: NEXT
310 PRINT: T=T+1: NEXT: NEXT
320 CLOSE 4: END
1000 DATA 13.75, .0587105751
2000 DATA "  NOOT    WAARDE"
3000 DATA "     HI    LO     HEX"
4000 DATA "C C#D D#E F F#G G#A A#B "
5000 DATA "0123456789ABCDEF"
```

(Illustratie 2)

Register $01. FREQ HI

Dit register bevat de Hi-byte van de frequentiewaarde. Zie verder de toelichting bij register $00.

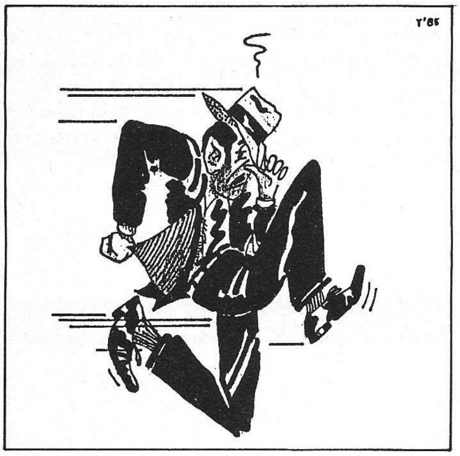
Register $02. PW LO (Pulse Width Lo-byte)

Als er gekozen is voor "Pulse" als Golfvorm (zie Register $04) bepaalt de waarde in dit en het volgende register de z.g.n. Pulse-breedte. De Pulse-breedte wordt uitgedrukt in een 12-bits waarde: de toestand van register $03's hoogste 4 bits (4-7) is hierbij niet van belang.

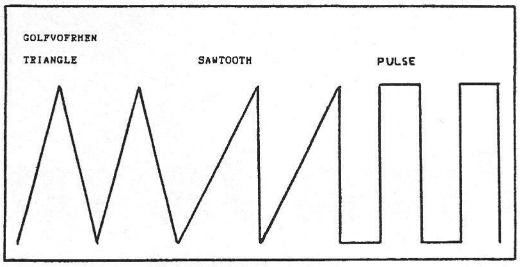
In de praktijk bepaalt de Pulse-breedte of een toon dun of vol zal klinken. De volheid neemt toe bij opklimmende waardes van $X000-$X800. Hij neemt weer af van $X800-$XFFF. Met andere woorden: bij waardes van $X000 en $XFFF is de volheid minimaal en bij de waarde $X800 is hij maximaal.

Register $03. PW HI (Pulse Width Hi-byte)

In dit register wordt de Hi-byte van de Pulse-breedte waarde geladen. Alleen de waarde van de laagste 4 bits (0-3) is hierbij van belang. Deze waarde kan $X0-$XF zijn.



GOLF-VORMEN



Register $04. CONTROL REGISTER

De bits van dit register hebben elk een eigen funktie.

Bit 0. GATE

Door bit 0 achtereenvolgens 1 en 0 te maken speel je een toon. In dit bit 1, dan begint er een A/D/S-patroon. Als het bit 0 wordt gemaakt volgt er een R-patroon. Deze termen worden bij de behandeling van registers $05 en $06 verklaard.

Bit 1. SYNC

Als dit bit 1 is, wordt de output van Stem 1 gesynchroniseerd met de output van stem 3. Hierbij is het gewenst dat de frequentie van Stem 3 lager is dan die van Stem 1. In het geval van Stem 2, wordt deze gesynchroniseerd met Stem 1. Stem 3 kan gesynchroniseerd worden met Stem 2. Is het SYNC-bit 0, dan worden de Stemmen normaal weergegeven. Het effekt van synchronisatie kun je het beste in de praktijk uitproberen.

Bit 2. RING MOD

Voor bit 2 geldt hetzelfde als voor bit 1, alleen worden de 2 Stemmen nu niet gesynchroniseerd, maar "geringmoduleerd". Ringmodulatie is uitsluitend te horen als de hoofdstem een driehoeksgolfvorm heeft (zie bits 4-7). Ook hier geldt: alleen door uitproberen kan het effekt duidelijk worden gemaakt.

Bit 3. TEST

Door dit bit 1 te maken reset je de betreffende stem (alle registers worden 0). Het bit wordt in de praktijk vrijwel niet gebruikt.

Bit 4. TRIANGLE (Driehoek)

Om diverse klankkleuren mogelijk te maken kun je per stem kiezen uit 4 verschillende Golfvormen (bits 4-7). Het betreffende bit dient dan 1 te worden gemaakt. Je kunt de Golfvormen ook kombineren, maar daar wordt de geluidskwaliteit niet beter door. Als bit 4 "1" is wordt de Driehoeksgolfvorm gekozen. Deze produceert een zachte, egale toon. Hij is ideaal voor het imiteren van fluitinstrumenten.

Bit 5. SAWTOOTH (Zaagtand)

De Zaagtand is rijk aan boventonen, en heeft een vol geluid. Je kunt hem gebruiken voor het imiteren van bijvoorbeeld violen en trompetten. Ook voor "synthesizergeluiden" is hij zeer geschikt.

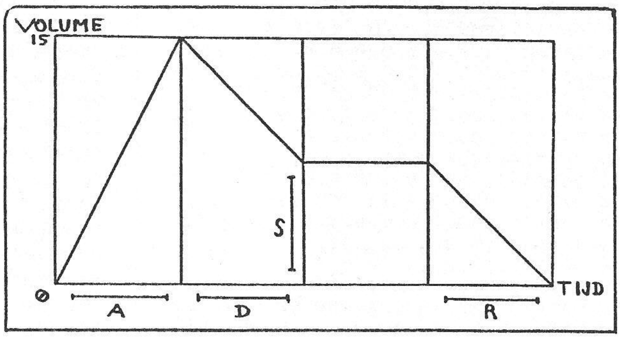
Bit 6. PULSE (Blokgolf)

Deze golfvorm geeft een strakke, harde toon. Door de Pulse-breedte te veranderen (zie register $02) kan deze toon een dunne tot een volle klank krijgen. Als je de Pulse-breedte verandert terwijl er een toon klinkt, kan dit een effekt veroorzaken wat "phasing" wordt genoemd.

Bit 7. NOISE (Ruis)

De Ruis-golfvorm dient voor het produceren van percussiegeluiden, zoals drums. Je kunt er geen melodie mee spelen.

A/D/S/R, de envelop



(Illustratie 3)

Met ADSR wordt de geluidsvorm bepaald, hoe snel het geluid aanzwelt en afneemt.

Register $05. ATTACK/DECAY
Dit register heeft twee funkties; het Hi-nybble (bits 4-7) bepaalt de Attack-waarde en het Lo-nybble (bits 0-3) de Decay-waarde. Deze waardes kunnen dus per nybble 0-15 zijn. De Attack-waarde heeft betrekking op de tijd die verstrijkt voordat een toon na aanzet zijn maximale volume bereikt. De Decay-waarde stelt de tijd voor die ligt tussen het "vallen" van het volume van maximaal naar een niveau dat aangegeven wordt met de Sustain-waarde. (zie register $06).
Bij verschillende typen instrumenten passen verschillende A/D-waardes. Een orgel bereikt bijvoorbeeld meteen na aanslag zijn maximale volume, dat verder konstant blijft. Attack en Decay zullen dus beide 0 moeten zijn. Het volume van een viool zal echter eerst aanzwellen voordat het maximale niveau wordt bereikt, waarna het zal dalen naar een niveau waarop de toon voor onbeperkte tijd kan worden aangehouden. Attack en Decay zullen dan bijvoorbeeld allebei 8 zijn.

Register $06. SUSTAIN/RELEASE
Ook register $06 is opgedeeld in twee nybbles met elk een eigen funktie. Het Hi-nybble (bits 4-7) bepaalt Sustain-niveau en het Lo-nybble (bits 0-3) de Release-waarde. Het Sustain-niveau komt neer op: het volumeniveau (0-15) waarop een toon wordt aangehouden nadat het A/D-patroon is afgelopen. De toon blijft op dit niveau net zo lang klinken to het GATE-bit van register $04 "0" wordt gemaakt. Vervolgens begint het Release-patroon. Hierbij daalt het Volume , in een tijdsbestek dat wordt aangegeven met de Release-waarde, van Sustain-niveau naar 0.

Bijgaande tabel geeft de tijdsduren aan die horen bij de A-, D- en R-waardes. De afkortingen MS en S staan hierbij voor milliseconden (1/1000 sec.) en seconden. De tabel is ontleend aan de Reference Guide, die zijn waardes baseert op een klokfrequentie van 1.0 Mhz. In Nederland werkt de 64 echter met 0.985 Mhz, zodat de werkelijke tijdsduren wat hoger uitvallen dan de tabel aangeeft.

WAARDE	ATTACK	DEC/REL
HEX/DEC	TIJD	TIJD
$00 0	2 MS	6 MS
$01 1	8 MS	24 MS
$02 2	16 MS	48 MS
$03 3	24 MS	72 MS
$04 4	38 MS	114 MS
$05 5	56 MS	168 MS
$06 6	68 MS	204 MS
$07 7	80 MS	240 MS
$08 8	100 MS	300 MS
$09 9	250 MS	750 MS
$0A 10	500 MS	1,5 S
$0B 11	800 MS	2,4 S
$0C 12	1 S	3 S
$0D 13	3 S	9 S
$0E 14	5 S	15 S
$0F 15	8 S	24 S

Het programmeren met A/D/S/R-patronen is zoals je ziet een vrij ingewikkelde zaak. De meeste (ook "professionele") programmeurs maken dan ook geen gebruik van deze mogelijkheid.

Melodisch zou er dan nog genoeg moois over kunnen blijven, maar zelfs daar is meestal geen sprake van. Schrijnende voorbeelden van deze muzikale armoede zijn bijvoorbeeld "Pitfall" en "Frogger". Juist door gebruik te maken van het A/D/S/R-patronen kan het "zeurende notenbrei-effekt" worden voorkomen. In dit verband zou "Ghostbusters" een lichtend voorbeeld moeten zijn voor allen die de computermuziek een warm hart toedragen.

FILTERS

Dan komen we nu toe aan de behandeling van de 8 gezamenlijke registers. De eerste 4 hebben betrekking op de filtering en het volume. We maken een sprong in de registernummering, want we slaan de individuele registers van stem 2 en 3 over.

Register $15. FC LO (Filter Freq Lo-byte)
Alleen de laagste 3 bits van dit register spelen een rol. Zij vormen tevens de laagste 3 bits van de Filterfrequentiewaarde. Elke toon heeft, zoals we zagen, een bepaalde frequentie. Maar afgezien van die grondfrequentie worden er ook nog boven- en ondertonen geproduceerd. De mate waarin die voorkomen bepaalt in hoge mate de uiteindelijke klankkleur van de toon. Deze klankkleur kun je regelen door de Golfvorm in te stellen.
SID heeft echter ook de mogelijkheid om alle (bij-) tonen boven of onder een bepaalde frequentie niet te laten horen. Dat gebeurt door middel van filtering. De Filterfrequentie geeft dus aan, vanaf waar er wordt gefilterd.
Filteren moet je meer zien als het "bijsturen" van de klankkleur, dan als het wezenlijk veranderen ervan. Het effekt komt meestal neer op het doffer, dan wel scherper maken van een toon.

Register $16. FC HI (Filter Freq HI-byte)
Dit register bevat de Hi-byte van de Filterfrequentie. De filterregisters kunnen waardes van $000-$7FF bevatten. Eigenaardig genoeg is het Lo-byte register 3-bits i.p.v. het Hi-byte register, zoals te doen gebruikelijk in dit soort gevallen. Let dus goed op bij het gebruik van deze registers.

Register $17. RES/FILT
Dit register is weer eens opgedeeld in een Hi- en een Lo-nybble. Het Hi-nybble bevat de Resonantie-waarde. Resonantie komt neer op het benadrukken van een bepaalde frequentie uit het toonbeeld. De waardes in de frequentieregisters bepalen hierbij welke frequentie resoneert. Er kan sprake zijn van geen Resonantie (0) tot maximale Resonantie (15).
De bits van het Lo-nybble hebben elk een eigen funktie. Ze bepalen of een toon ongefilterd (0) dan wel gefilterd (1) uit de luidsprekers komt. Bit 0 is voor Stem 1, bit 1 voor Stem 2 en bit 2 voor stem 3. Bit 3 heeft dezelfde funktie voor een externe stem, waarvoor de mogelijkheid bestaat om deze via pin 5 van de audio/video-plug de computer binnen de laten komen.
Je kunt dus wel voor elke Stem apart bepalen of deze al dan niet gefilterd wordt, maar niet op welke frequentie dit gebeurt. Die waarde is voor alle Stemmen gelijk.

Register $18 MODE/VOL
De nybbles van dit register hebben weer elk een eigen taak. Het Hi-nybble dient voornamelijk om de Filter-mode te bepa-

len. De funkties van de afzonderlijke bits zijn:

○ Bit 4. LP (Low Pass)
Als dit bit 1 is worden alle frequenties boven de waarde in de Filterfrequentie-registers weggefilterd. Het resultaat is meestal een wat dof geluid.

○ Bit 5. BP (Band Pass)
Als dit bit 1 is worden alle frequenties onder en boven de Filterwaarde weggefilterd. Je krijgt hierdoor een dunne, fragiele klank.

○ Bit 6. HP (High Pass)
Is dit bit 1, dan worden alle frequenties onder de Filterwaarde weggefilterd. Dit heeft als gevolg een scherp, helder geluid. De diverse Filter-modes kun je ook kombineren. Maar behalve voor speciale effekten is dit niet aan te raden. De geluidskwalitiet wordt er vrijwel altijd slechter op. ○ Bit 7. 3 OFF (Stem 3 uit)
Als bit 7 "1" is, dan is Stem 3 niet meer hoorbaar. Dat is bijvoorbeeld handig bij Ringmodulatie van Stem 1 met Stem 3, waarbij Stem 3 alleen een funktie heeft als producent van een Modulatiefrequentie.

Het Lo-nybble van register $18 dient om het Volume-niveau (0-15) te bepalen. Ook dit is voor alle Stemmen gelijk. Maar in sommige gevallen is het mogelijk om met gebruikmaking van de Sustain-waardes verschillen per Stem te bewerkstelligen.

OUTPUT

De tot nu toe behandelde registers van de SID-chip zijn allemaal Write-only. Dat wil zeggen dat je er alleen maar naar toe kunt schrijven (=waardes erin zetten). Bij lezen (=inhoud opvragen) komen de gelezen waardes niet overeen met de werkelijke inhoud. Voor diverse doeleinden is het echter wel eens handig om ook wat output van SID te krijgen. Registers $19-$1C voorzien in deze behoefte. Ze zijn allen Read-only (schrijven heeft geen effekt).

Het programmeren met deze registers is echt iets voor de specialisten; vandaar dat we er niet te lang bij stil zullen staan. Register $19 bevat de weerstandswaarde van Potentiometer X. Register $1A bevat de weerstandswaarde van Potentiometer Y. Register $1B bevat de hoogste 8 output-bits van Oscillator 3, die gerelateerd is aan de Golfvorm van Stem 3. Bij Ruis als Golfvorm produceert dit register willekeurige waardes van 0-255. Register $1C is gerelateerd aan het A/D/S/R-patroon van Stem 3.

De volgende keer is de beeld-chip VIC het onderwerp van weer een aflevering "Programmeren in machinetaal".

S.B.

```
10 A=36864: FOR X=0 TO 297
20 READ Y: POKE A+X,Y: NEXT
30 SYS A+121: END
100 DATA43,221,0,0,0,0,39,20
110 DATA36,226,0,0,32,220,29,70
120 DATA27,162,0,0,29,166,0,0
130 DATA27,162,0,0,0,0,0,0
140 DATA36,226,0,0,0,0,39,20
150 DATA43,221,0,0,39,20,36,226
160 DATA32,220,0,0,29,70,0,0
170 DATA32,220,0,0,0,0,0,0
180 DATA29,70,0,0,29,70,0,0
190 DATA0,0,0,0,19,138,0,0
200 DATA21,238,0,0,24,158,0,0
210 DATA21,238,0,0,0,0,0,0
220 DATA1,0,0,8,0,64,240,0
230 DATA0,0,8,0,66,64,0,0
240 DATA0,8,0,2,32,0,0,0
250 DATA15,120,162,24,189,96,144,157
260 DATA0,212,202,16,247,162,95,189
270 DATA0,144,157,192,145,157,32,146
280 DATA157,224,146,157,64,147,202,16
290 DATA238,169,0,133,251,133,252,162
300 DATA0,160,1,189,0,144,240,3
310 DATA153,0,212,189,0,146,240,3
320 DATA153,7,212,189,0,147,240,3
330 DATA153,14,212,232,136,16,228,189
340 DATA254,143,240,5,169,17,141,4
350 DATA212,165,251,240,10,189,254,145
360 DATA240,5,169,33,141,11,212,165
370 DATA252,240,10,189,254,146,240,5
380 DATA169,65,141,18,212,169,64,32
390 DATA31,145,189,0,144,240,5,169
400 DATA16,141,4,212,189,0,146,240
410 DATA5,169,32,141,11,212,189,0
420 DATA147,240,5,169,64,141,18,212
430 DATA169,64,32,31,145,224,32,208
440 DATA2,132,251,224,64,208,2,132
450 DATA252,224,96,208,132,240,128,133
460 DATA2,136,208,253,198,2,208,249
470 DATA136,96
```

MT PROGRAMMA

Het programma van deze aflevering speelt de 3-stemmige canon "Viva la musica" tot in het oneindige. De routine Copy kopieert de noten-data van $9000-$905F voor Stem 2 en 3 naar vanaf $9200 en $9300, en vult vervolgens SID met vaste waardes. Freq zet freqentiewaardes in de betreffende registers. Speel zet een A/D/S-patroon in en roept vervolgens de routine Wacht aan. Daarna begint een R-patroon, weer gevolgd door Wacht en er wordt voor de volgene noot terug naar Freq gesprongen. Dit herhaalt zich tot de computer gereset wordt.

```
9000 2B DD 00 00 00 00 27 14
9008 24 E2 00 00 20 DC 1D 46
9010 1B A2 00 00 1D 46 00 00
9018 1B A2 00 00 00 00 00 00
9020 24 E2 00 00 00 00 27 14
9028 2B DD 00 00 27 14 24 E2
9030 20 DC 00 00 1D 46 00 00
9038 20 DC 00 00 00 00 00 00
9040 1D 46 00 00 1D 46 00 00
9048 00 00 00 00 13 8A 00 00
9050 15 EE 00 00 18 9E 00 00
9058 15 EE 00 00 00 00 00 00
9060 01 00 00 08 00 40 F0 00
9068 00 00 08 00 42 40 00 00
9070 00 08 00 02 20 00 00 00
9078 0F
```

```
9079 78           SEI
907A A2 18        LDX #$18
907C BD 60 90     LDA $9060,X
907F 9D 00 D4     STA $D400,X
9082 CA           DEX
9083 10 F7        BPL $907C
9085 A2 5F        LDX #$5F
9087 BD 00 90     LDA $9000,X     Copy
908A 9D C0 91     STA $91C0,X
908D 9D 20 92     STA $9220,X
9090 9D E0 92     STA $92E0,X
9093 9D 40 93     STA $9340,X
9096 CA           DEX
9097 10 EE        BPL $9087
9099 A9 00        LDA #$00
909B 85 FB        STA $FB
909D 85 FC        STA $FC
909F A2 00        LDX #$00
90A1 A0 01        LDY #$01
90A3 BD 00 90     LDA $9000,X
90A6 F0 03        BEQ $90AB
90A8 99 00 D4     STA $D400,Y
90AB BD 00 92     LDA $9200,X     Freq.
90AE F0 03        BEQ $90B3
90B0 99 07 D4     STA $D407,Y
90B3 BD 00 93     LDA $9300,X
90B6 F0 03        BEQ $90BB
90B8 99 0E D4     STA $D40E,Y
90BB E8           INX
90BC 88           DEY
90BD 10 E4        BPL $90A3
```

```
90BF BD FE 8F     LDA $8FFE,X
90C2 F0 05        BEQ $90C9
90C4 A9 11        LDA #$11
90C6 8D 04 D4     STA $D404
90C9 A5 FB        LDA $FB
90CB F0 0A        BEQ $90D7
90CD BD FE 91     LDA $91FE,X
90D0 F0 05        BEQ $90D7
90D2 A9 21        LDA #$21
90D4 8D 0B D4     STA $D40B
90D7 A5 FC  A/D/S LDA $FC
90D9 F0 0A        BEQ $90E5
90DB BD FE 92     LDA $92FE,X
90DE F0 05        BEQ $90E5
90E0 A9 41        LDA #$41
90E2 8D 12 D4     STA $D412
90E5 A9 40        LDA #$40
90E7 20 1F 91     JSR $911F
90EA BD 00 90     LDA $9000,X
90ED F0 05        BEQ $90F4
90EF A9 10        LDA #$10        Speel
90F1 8D 04 D4     STA $D404
90F4 BD 00 92     LDA $9200,X
90F7 F0 05        BEQ $90FE
90F9 A9 20   R    LDA #$20
90FB 8D 0B D4     STA $D40B
90FE BD 00 93     LDA $9300,X
9101 F0 05        BEQ $9108
9103 A9 40        LDA #$40
9105 8D 12 D4     STA $D412
9108 A9 40        LDA #$40
910A 20 1F 91     JSR $911F
910D E0 20        CPX #$20
910F D0 02        BNE $9113
9111 84 FB        STY $FB
9113 E0 40        CPX #$40
9115 D0 02  Begin BNE $9119
9117 84 FC    ?   STY $FC
9119 E0 60        CPX #$60
911B D0 84        BNE $90A1
911D F0 80        BEQ $909F
911F 85 02        STA $02
9121 88           DEY
9122 D0 FD        BNE $9121
9124 C6 02        DEC $02         Wacht
9126 D0 F9        BNE $9121
9128 88           DEY
9129 60           RTS
```

Wie zelf zijn programma's schrijft, kan heel wat aftobben met het verkrijgen van antwoorden van de (eind) gebruiker, ofwel de invoer vanaf het toetsenbord. Nooit meegemaakt dat, na uren programmeren, een overijverige buurman door het indrukken van een paar toetsen een 'Extra Ignored' of 'Redo from start' veroorzaakt en daarmee de hele scherm lay-out ruineert en mogelijk ook het programma onderbreekt? In een ander geval zou een moment van onoplettendheid, het invoeren van cijfers in plaats van letters of omgekeerd, het programma abrupt kunnen beeindigen en de reeds ingevoerde data verloren doen gaan.

INVOER

Als je in je programma's veel invoer te verwerken hebt en ervoor wilt zorgen dat de syntax van de invoer altijd juist is, zul je je afvragen of de wijze van invoerbehandeling, zoals we die kennen van de betere professionele software, ook mogelijk is in simpel alledaags Basic.

Invoerroutine

Op het eerste gezicht lijkt dit een bewerkelijke klus. Als bij iedere invoer het verkregen antwoord moet worden gecontroleerd, kan dit heel wat Basicregels extra opleveren. Hierdoor wordt niet alleen het programma langer (zowel in het computergeheugen als op disk of cassette), maar ook langzamer en onoverzichtelijker. Een logische oplossing is alle invoer onder te brengen in een subroutine.

Input en Get

Een tweede probleem is de werking van het Input-commando. Hierbij is het onmogelijk om iedere ingedrukte toets te controleren. Slechts na het beeindigen van de invoer, bij het indrukken van de Returntoets kan bekeken worden of de invoer correct is. Is deze niet correct dan zal opnieuw om invoer moeten worden gevraagd, wat niet alleen vrij bewerkelijk is voor de programmeur, maar bovendien ergenis opwekt bij de gebruiker. Anders ligt het bij het Get-commando, dat steeds een karakter van het toetsenbord haalt en het daardoor mogelijk maakt iedere ingedrukte toets op juistheid te controleren en deze al dan niet te accepteren. Wel moet bij het gebruik van het Get-commando voor een kunstmatige cursor worden gezorgd. Dit kan gedaan worden voor toepassing van de eerste "cursortruc" die te vinden is op bladzijde 17 van Commodore-Info, nummer 2, 1985. Een andere mogelijkheid (minder bewerkelijk) is het gebruik van een willekeurig karakter in plaats van een knipperende cursor. Dit kan een "reverse-spatie" zijn, het grafische teken "COMM-P" of een ander karakter.

Controle

Een volgende stap is de methode waarop de ingevoerde karakters worden gecontroleerd. Voor één type invoer op een bepaalde plaats in een programma is dit niet zo'n probleem. Bij een uniforme subroutine wordt het wat ingewikkelder. Bepaalde antwoorden mogen alleen bestaan uit cijfers, andere alleen uit letters en weer andere uit letters en cijfers samen, maar geen lees- of grafische tekens. Daarom is het allereerst zaak alle karakters onder te verdelen in een aantal groepen, zodat later bij iedere invoer aangegeven kan worden welke groep "legaal" is, en dus geaccepteerd mag worden en welke groep "illegaal".

Lengte van de invoer

Ook moet de maximale lengte van de invoer in de gaten worden gehouden. Het is tenslotte niet de bedoeling dat de gebruiker met zijn invoer het hele scherm volschrijft, terwijl het antwoord bij voorbeeld uit slechts 16 karakters had mogen bestaan. De lengte van een invoer is eenvoudig bij te houden door het tellen van het aantal legale toetsen, waarvan wordt afgetrokken het aantal karakters dat met de Delete-toets wordt gewist. Voordat het programma de invoerroutine kan uitvoeren, moet opgegeven worden wat de maximale lengte van de invoer is (variable IL). De programmeur kan hierbij kiezen of de invoer een maximale lengte heeft of een exacte lengte (niet langer, maar ook niet korter). De invoerroutine, die bij dit artikel is afgebeeld, dient in het laatste geval (exacte lengte) als volgt te worden gewijzigd:
158 IF (12=13 OR 12=141) AND 13=IL THEN 168.

Selectie van karakters

Naaste de 'vaste' groepen kan ook een selectie van karakters worden gemaakt om in de routine te gebruiken. Hiervoor wordt de array IS$(n) gebruikt, waarbij het nummer van de selectie wordt aangegeven door de variabele IS. Bijvoorbeeld: bij een invoer waar een hexadecimaal getal gevraagd wordt, heb je, als legale invoer, de cijfers 0 t/m 9 en de letters a t/m z nodig. Zonder verder iets aan de routine te veranderen kun je dit probleem oplossen door de waarde 129 (1+128) te geven en de array IS$ (0) als volgt te definieren:
IS = 0 : IS$ (IS) = "ABCDEF"

Door IS een andere waarde te geven kunnen meerdere selecties naast elkaar worden gebruikt. Is het aantal groter dan 10 dan zal de array IS$(n) eerst moeten worden worden gedimensioneerd.

Invoersoorten

Er zijn een aantal soorten invoer, die geregeld terugkomen. Soms willen we mogelijke invoer beperken tot bijvoorbeeld alleen letters of cijfers. Een voorbeeld van de indeling, die daarvoor nodig is naar invoertypen is de volgende:

❶ De cijfers 0 t/m 9 (met eventueel het minteken)	I=1
❷ De letters a t/m z (zonder gebruik van Shift- of Comm. logo-toets)	I=2
❸ De toetsen A t/m Z, ingedrukt samen met de Shift- toets	I=4
❹ Alle leestekens e.d. behalve het aanhalingsteken (")	I=8
❺ Alle grafische tekens, met de Comm. logo-toets verkregen	I=16
❻ De functietoetsen	I=32
❼ Spatie	I=64
❽ Een speciale selectie (afhandelijk van programma)	I=128

De waarde 'I' geeft in een programma aan welke groep legaal is en welke niet. Afhankelijk van deze waarde zal een bepaalde toets al dan niet worden geaccepteerd. De toekenning van de waarde van I in bovenstaand overzicht is niet zonder reden zo gekozen. Op deze manier kan een willekeurige combinatie van enkele groepen worden gemaakt, waarbij I wordt verkregen door optelling van de in het overzicht vermelde waarden. I kan daarna vergeleken worden met acht bits uit een byte, waarvan iedere bit afzonderlijk is te testen. Wanneer alle toetsen gebruikt mogen worden krijgt I de waarde 255. Een voorbeeld: bij een invoer mogen alleen de cijfers (groep 1) en de lettertoetsen A t/m Z (groep 3) geaccepteerd worden. I wordt dan 5 (1+4). In de invoerroutine wordt vervolgens nagegaan of "I and I' gelijk is aan 0. Is dit niet het geval (zoals in dit voorbeeld) dan worden de cijfertoetsen (groep 1) geaccepteerd. Vervolgens wordt gekeken of '1 and 2' gelijk is aan 0. Dit is hier

```
100   REM --- BYTES & CO ---
101   REM ---   INVOER   ---
102 PRINT"▮ ▬▮";:I$="":I3=0:IFI=0ORIL=0THEN168
104 POKE198,0:WAIT198,1:GETI1$:I2=ASC(I1$)
106 IFI3=ILTHEN158
108 IF(IAND1)=0THEN114
110   IF(I2>47ANDI2<58)THEN164
112   IF(I3=0ANDI2=45)THEN164
114 IF(IAND2)=0THEN118
116   IF(I2>64ANDI2<91)THEN164
118 IF(IAND4)=0THEN124
120   IF(I2>97ANDI2<122)THEN164
122   IF(I2>192ANDI2<219)THEN164
124 IF(IAND8)=0THEN136
126   IFI2=34THEN166
128   IF(I2>32ANDI2<48)THEN164
130   IF(I2>57ANDI2<65)THEN164
132   IF(I2>90ANDI2<96)ORI2=192THEN164
134   IF(I2=126ORI2=255)THEN164
136 IF(IAND16)=0THEN142
138   IF(I2>160ANDI2<192)THEN164
140   IF(I2>223ANDI2<255)THEN164
142 IF(IAND32)=0THEN146
144   IF(I2>132ANDI2<141)THEN164
146 IF(IAND64)=0THEN150
148   IFI2=32ORI2=160ORI2=224THEN164
150 IF(IAND128)=0ORIS$(IS)=""THEN158
152   FORI4=1TOLEN(IS$(IS))
154     IFI2=ASC(MID$(IS$(IS),I4,1))THEN164
156   NEXT
158 IFI2=13ORI2=141THEN168
160 IFI2=20THENIFI3>0THENI3=I3-1:I$=LEFT$(I$,
    LEN(I$)-1):PRINT" ▮▮▮ ▬▮";
162   I1$="":GOTO166
164 I3=I3+1:I$=I$+I1$:PRINTI1$;"▮ ▬▮";
166 GOTO104
168 PRINT" ":RETURN
199
```

waar, zodat de lettertoetsen a t/m z (groep 2) niet worden geaccepteerd. Daarna wordt bekeken of groep 3 (lettertoetsen A t/m Z met Shift-toets ingedrukt) legaal is door na te gaan of 'I and 4' al dan niet gelijk is aan 0. Zo kan voor ieder ingevoerd karakter worden gecontroleerd of deze deel uitmaakt van een vooraf opgegeven legale groep. Is dit het geval dan wordt de invoer geaccepteerd, en zo niet dan wordt deze geignoreerd. Naast de hierboven genoemde toetsen dienen nog twee toetsen geaccepteerd te worden. Dit zijn de Delete-toets (zonder Shift) om het laatst ingevoerde karakter te wissen en de Return toets om de invoer te beeindigen. De Delete-toets mag trouwens niet worden gebruikt en dus ook niet worden geaccepteerd als er nog geen invoer op het scherm staat.

Gebruik van de routine

Het resultaat van de routine (het antwoord) is de string I$.
Neem de routine apart op op cassette of diskette en reserveer in al je programma's de regels 100 t/m 199 voor deze routine.
Hierdoor kun je met behulp van een APPEND- of MERGE-programma deze routine aan je eigen programma toevoegen, zodat je het geheel slechts eenmaal hoeft in te tikken.
In plaats van 'PRINT"(RVS) (SPATIE) (RVS OFF) (CRSR LINKS)",in regel 102 en 164 kan ook worden gebruikt 'PRINT" (KARAKTER) (CRSR LINKS)", en kan ook de print-instructie in regel 160 op een overeenkomstige wijze worden gewijzigd. Ook kan op deze plaats de eerder genoemde 'cursortruc' worden ingevoegd (door middel van een afzonderlijke subroutine).
Het testen van de ASCII-waarde van de ingedrukte toets kan naar believen worden aangepast, zodat een eigen groepsindeling van de toetsen mogelijk wordt. Gebruik hiervoor de tabel 'ASCII and CHR$ codes' uit de handleiding van je computer.

Overzicht van gebruikte variabelen

De variabelen kunnen in twee groepen worden verdeeld. De eerste groep bestaat uit de variabelen waaraan door het programma (de programmeur) een bepaalde waarde worden gegeven en de tweede groep bevat de variabelen die door de invoerroutine worden gebruikt. De eerste groep moet, vooraf, door de programmeur worden ingevuld.

```
10 REM ---  BYTES & CO  ---
11 REM --- INVOER  TEST ---
20 I        =  255  : REM  KEUZE 'LEGALE' GROEPEN
30 IL       =   10  : REM  (MAXIMALE) LENGTE INVOER
40 IS       =    1  : REM  NUMMER VAN SELECTIE
50 IS$(IS) = "ABC" : REM  SELECTIE KARAKTERS
60
70 PRINT"■":GOSUB100:PRINT"■";I$:END
80
90 REM --- GROEPEN  (I) ---
91 I =   1 : REM"  0 T/M 9 EN -
92 I =   2 : REM"  A T/M Z
93 I =   4 : REM"  ♠ T/M ♦
94 I =   8 : REM"  LEESTEKENS E.D. BEHALVE " ·
95 I =  16 : REM"  ALLE TOETSEN MET ┤┌\\. INGEDRUKT
96 I =  32 : REM"  FUNCTIETOETSEN
97 I =  64 : REM"  SPATIE EN SHIFT-SPATIE
98 I = 128 : REM"  SPECIALE SELECTIE
99
```

Groep 1:
I aanduiding legale groepen (zie overzicht bij 'Controle')
IL (maximale) lengte I$
IS$(N) selectie van speciale karakters
IS nummer van toe te passen selectie

Groep 2:
I$ resultaat van de totale invoer
I1$ ingedrukte toets
12 ASCII- waarde I1$
13 actueel aantal legale karakters
14 teller bij selectie

Bij het programmeren zijn er een aantal dingen, waar je op moet letten. Dat je bv. met 10 omhoog gaat bij het nummeren, de subroutines achteraan zet en hier en daar wat REM's zet, weet vrijwel iedereen. Maar er is meer, en Peter Cremer geeft wat aanwijzingen.

Programmer's Tips & Truc's

Door Peter Cremer.

Het gewone gezonde verstand verlaat vele programmeurs, zogauw ze achter de toetsen zitten. Dan denken ze alleen nog maar in routines, terwijl de gebruiker van hun programma vergeten raakt.

Tip 1. Wees vriendelijk.

Je hebt het vast wel eens meegemaakt. Je laadt en runt een programma.Lange tijd zie je niets. Dit wordt meestal veroorzaakt doordat een programma al die tijd bezig is met het lezen van DATA, of omdat er een niet zo'n snelle sorteerroutine in het programma zit.
Maak je zelf ook programma's met lange wachttijden (5 seconden en meer), wees dan gebruikersvriendelijk en laat de gebruiker zien dat hij even geduld moet hebben. Het kost je weinig moeite om een dergelijke PRINT opdracht in je programma aan te brengen.

Tip 2. Zorg voor snelheid.

Probeer altijd je programma zo snel mogelijk te houden. Dit kun je bereiken met verschillende technieken.

❶ Als je in je programma subroutines gebruikt (GOSUB), zet deze dan altijd bovenin je programma. Staat er ergens bijvoorbeeld GOSUB 1000, dan gaat de computer niet rechtstreeks naar regel 1000, maar begint bovenin net zo lang te zoeken tot hij die regel vindt.
❷ Gebruik je in je programma complexe berekeningen, dan kun je deze het beste vastleggen d.m.v. het DEF statement.Ook dit doe je aan het begin van je programma.
❸ Probeer zo veel mogelijk opdrachten op een regel te zetten.Bij een wat groter programma scheelt dat aardig in het geheugengebruik en dus in snelheid.
❹ Er zijn nog meer mogelijkheden om de snelheid op te voeren.

Opdracht: Ga zelf eens na wat sneller is om een scherm schoon te maken:
a.PRINT "CLR/HOME".
b.SYS 58692.
c.PRINTCHR$(147).
Je zult zien dat er een bij zit die 7 tot 9 maal sneller werkt dan de anderen. Ga ook het snelheidsverschil eens na tussen onderstaande one-liners die beide hetzelfde doen.
O for I= 1 to 1000: PRINT I:NEXT en O I=I+1:PRINT I: IF<1000 THEN O. ●

Truc 1

Beschermde Invoer.

Als je binnen een programma werkt met INPUT, dan kan er van alles misgaan. Vraag je om numerieke invoer en de gebruiker voert alpha-numerieke invoer in, dan kun je er donder op zeggen dat het programma zegt: REDO FROM START. Een ingevoerde komma levert de mededeling op: extra ignored. Al deze narigheid kun je omzeilen door gebruik te maken van de GET instruktie. Rustig maar, ik hoor je al zeggen:"daar kan ik maar een teken in kwijt". Onderstaande subroutine beschermd je invoer. Je kunt deze routine makkelijk uitbreiden met enkele test op ongewenste tekens . Nadeel van de GET instrukties is wel, dat je geen cursor in beeld krijgt. Wil je die er toch bij hebben, gebruik dan truc 2. Succes.
Deze routine verplicht de gebruiker te antwoorden met invoer. Een kale RETURN wordt niet als geldige invoer beschouwd. Het resultaat van de invoer wordt opgeslagen in Q$. Het is eenvoudig om deze routine alleen geschikt te maken voor numerieke invoer. Een kwestie van een extra testje. Vergeet dan niet om door middel van bijvoorbeeld Q=VAL(QS) de zaak voor RETURN om te zetten.

```
1) Q$="":K$=""                         maak variabelen schoon
2) GETK$:IFK$="" THEN2                 wacht op invoer
3) IFK$=CHR$(13)THEN6                  test op RETURN
4) IFK$=CHR$(20) THEN8                 test op DEL
5) QS=QS+K$:PRINTK$;:GOTO2
                  voeg K$ toe aan Q$;haal nieuw teken
6) IFQS=""THEN2                        Q$ leeg? dan opnieuw
7) RETURN                              einde routine
8) IFQS=""THEN2                        Q$ leeg? dan opnieuw
9) QS+LEFT$(QS,LEN(QS)-1):PRINTK$;:GOTO2
               haal laatste teken van Q$ weg;haal nieuw teken
```

Truc 2.

Cursor in beeld bij GET instruktie.

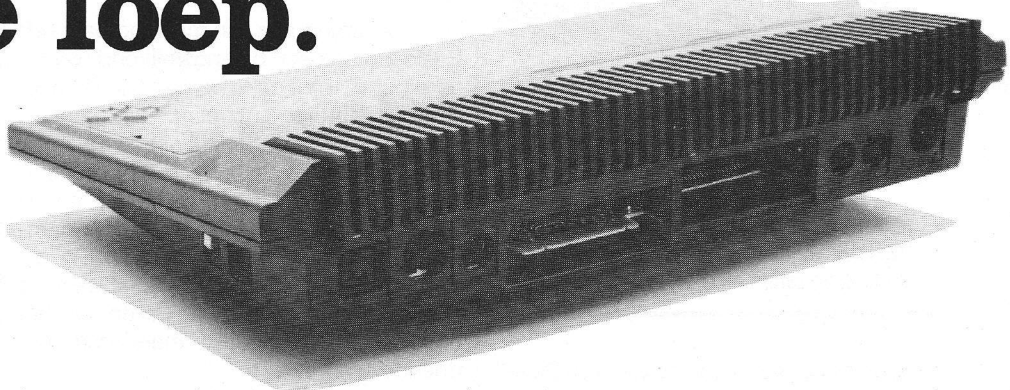
Dit is wel een zeer eenvoudige truc, maar bij velen onbekend.

```
1) POKE 204,0                          cursor aan
2) GETK$:IFK$=""THEN2                  wacht op invoer
3)IFPEEK(207)THEN3                     test blinkfase cursor
4)POKE 204,1                           cursor uit
5) PRINT K$
```

Gebruik je deze truc, dan wil je meestal eerst de gebruiker vertellen wat je van hem verwacht.Doe dat bijvoorbeeld zo:PRINT 'NOG EENS (J/N)"; Daarna pas je de truc toe. Je zult zien dat de cursor keurig achter deze regel staat te knipperen. (P.S. goed opletten dat je de punt-komma achter je PRINT opdracht niet vergeet. Anders komt de cursor een regel lager!)

Hebben ze een "dode" machine, afgeschilderd als winkeldochter, zijn de kopers van een Plus/4 echt de pineut? Of valt het allemaal wel mee en is de Plus/4 toch de geniale gebruiksmachine, die Jack Tramiel destijds zo trots aankondigde.

De Plus/4 onder de loep.



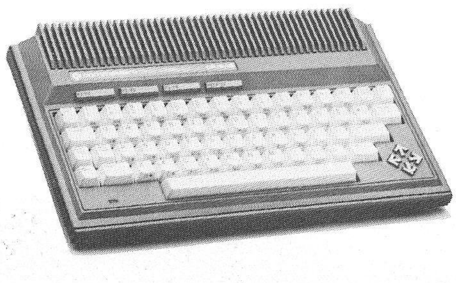
Op de eerste pagina van de handleiding van de Plus/4 laat Commodore ons weten, dat we een wijze beslissing hebben genomen, door deze nieuwe Plus/4 computer aan te schaffen. Want, zo staat er, de Plus/4 is de eerste huiscomputer, helemaal ontworpen voor productieve toepassingen. Natuurlijk kun je er ook alle dingen mee doen waarvoor je een huiscomputer meestal gebruikt, zoals programma's, spelletjes, simulaties en noem maar op. Maar het accent van deze machine ligt vooral op de serieuze toepassingen.

In ieder geval blijkt hieruit wel, dat het om een aardige computer moet gaan en dat klopt ook wel. Leuk van ontwerp, goed toetsenbord en veel meer mogelijkheden dan op het eerste gezicht misschien worden verwacht.
Standaard zitten er in de Plus/4 de nodige **Personal Productivity** pakketten, waarmee meteen kan worden gewerkt. Een tekstverwerker, spread-sheet en database zitten vast ingebakken in de Rom's van de Plus/4.
Hoewel er nogal wat negatieve opmerkingen over deze standaard software zijn gemaakt, betekent dit beslist niet, dat er niet met deze toepassingen kan worden gewerkt. Voor de 'huis/tuin/keuken en winkeltje' toepassingen doet deze Plus/4 software niet onder voor de losse pakketten die leverbaar zijn voor andere apparaten, zoals C-64 of MSX. Bovendien zijn de gegevens van de drie pakketten onderling uit te wisselen, zodat je werkelijk een handig systeem tot je beschikking hebt om je eigen gegevens in op te bergen en te verwerken. En daarbij maakt het niet uit of het nu bedragen zijn waarmee moet worden gerekend, of NAW-gegevens die nodig zijn om een in de tekstverwerker geschreven brief naar het juiste adres te versturen.

De C-16 en de Plus-4 zijn qua prijs/ prestatie topprodukten (geworden).

Het is daarom nogal vreemd dat er in het algemeen zo weinig animo voor de Plus/4 en z'n kleinere broer, de C-16 bestaat, waardoor deze machines de kans lopen, als de grote mislukking van Commodore de geschiedenis in te gaan. Overigens zijn er al meer dan tienduizend C-16's in ons land verkocht en dat is langzamerhand al een behoorlijke basis, reden waarom we in Commodore-Info ook meer aandacht aan deze machines gaan besteden. Toch is de handel vrij negatief, schuift ze als starterpakketten over de toonbank en wil er verder niet veel van weten. Ook Commodore zelf steunt deze machines in ieder geval niet meer van ganser harte. Toch is dat een miskenning van deze prima computers.
De direkte aanleiding voor deze geringe interesse moeten we waarschijnlijk zoeken in de richting van de hoeveelheid beschikbare externe software voor de 64 en daarmee de aarzeling van softwarehuizen om iets te schrijven voor de machines. Bovendien werden bij de aankondiging van eerst de 264 en 364 (de voorlopers van de Plus/ 4) nogal wat tactische fouten gemaakt en toen het ding er eindelijk was, kwam ook de 128 al om de hoek kijken. Er is in verhouding met b.v. de C-64 betrekkelijk weinig programmatuur op de markt, die geschikt is voor de C-16 & Plus/4. Maar dat betekent niet, dat er in 't geheel niets zou zijn. Denk bijvoorbeeld maar eens aan LOGO, dat voor de Plus/4 op cartridge verkrijgbaar is, en met de 64 KB MCT RAM uitbreiding ook prima werkt op de C-16. Zo zijn er nog wel een paar sterke softwaretools en -programma's op te noemen, die meehelpen de Plus/4 tot een volwaardige computer te maken. En zelfs zonder die zaken is het op zich gewoon een leuk ding, niet duur en voor bv. scholieren bijna ideaal.
Wat is er dan zo bijzonder aan deze Plus/4, die door de meesten van ons in vergelijking met de C-64 wordt beschouwd als een soort veredelde Vic-20. Om te beginnen beschikt de Plus/4 over 64K RAM, waarvan we voor Basic maar liefst 60K kunnen gebruiken. En dat zijn 28000 bytes meer dan bij de C-64! Verder zitten er 4 aparte cursortoetsen op het toetsenbord en zijn de funktietoetsen alvast voorgeprogrammeerd met de meest voorkomende opdrachten. Het is natuurlijk mogelijk om zelf een andere funktie aan deze toetsen toe te kennen. In plaats van de 16 bekende C-64 kleuren, hebben we nu de beschikking over 121 verschillende kleurschakeringen. Het plotten van grafische tekens in hires kan door Basic- opdrachten worden gedaan, dus geen peeks en pokes meer, en dat geldt ook voor de te produceren geluiden, waarvoor er zelfs twee toon generators zitten ingebouwd. Het scherm kan in diverse 'modes' worden aangestuurd, waaronder één, die ervoor zorgt

dat in 'split-screen' zowel tekst als hires kan worden getoond. Kom er maar eens om bij de C-64. Alle karakters en letters kunnen in 'reverse' op het scherm worden gezet, maar daarnaast ook nog eens aan



en uit flitsend, 'flashing', zodat het mogelijk wordt een regel tekst in het oog te laten springen.

Basic 3.5 lijkt erg op de Basic 7.0 van de 128, ze tellen bij CBM nogal digitaal.

De Basic interpreter van de C-16 en Plus/4 heeft als versienummer 3.5 mee gekregen, en is dan ook aanmerkelijk uitgebreider dan de Basic 2.0 waarmee de C-64 is uitgerust. Er zitten meer dan 75 echte Basic-commando's in. Een deel van de uitbreidingen wordt gebruikt om een gemakkelijke communicatie met de diskdrive tot stand te brengen. We kunnen nu b.v. de directory van een disk direkt op het scherm laten verschijnen, zonder dat het programma dat in het geheugen zit daardoor wordt verwijderd. Ook het laden, saven en kopiëren en 'backup'pen kan door het geven van één Basic opdracht gebeuren. Voor grotere machines was dit nooit een probleem, maar de meeste huiscomputers kennen deze mogelijkheden niet. Basic 3.5 beschikt over een uitgebreide toolkit, waarmee we in staat worden gesteld, op een vlotte manier onze eigen Basic programma's te schrijven en debuggen. Het automatisch nummeren en hernummeren van programma-regels is geen probleem voor de Plus/4 terwijl hele programma-delen -desgewenst- kunnen worden verwijderd. HELP, TRON en PRINT USING zijn ook het vermelden waard.

Voor de liefhebbers van machinetaal heeft Basic 3.5 ook een belangwekkend nieuwtje in petto, want deze versie beschikt over een uitgebreide machinetaal monitor, **TEDMON**, waarmee we naast de bekende zaken als dis- en assembleren ook machinetaalprogramma's kunnen saven en laden. Verder beschikt TEDMON over commando's als HUNT, FILL, MEMORY en TRANSFER. Om TEDMON vanuit Basic te kunnen benaderen, is het geven van het commando 'MONITOR' meer dan voldoende.

De grafische eigenschappen, die de C-16 en Plus/4 ons aanbieden liegen er evenmin om. Met de opdracht COLOR geven we scherm, tekens of achtergrond gemakkelijk een van de 121 beschikbare kleuren. Om te werken met high-resolution kun je volstaan met het veranderen van de scherm-mode (GRAPHIC) waarna je kunt beginnen met het tekenen van lijnen (DRAW), rechthoeken (BOX) of cirkels (CIRCLE), die door de mee te geven getallen allerlei vormen kunnen krijgen. Met PAINT bereik je dat een bepaald deel van het scherm wordt 'geverfd' in de door jou gewenste kleur. SSHAPE en GSHAPE zorgen er tenslotte voor dat gedeeltes van het scherm (in hires) kunnen worden toegekend aan gewone Basic string-variabelen en als zodanig kunnen worden weggezet naar disk of cassette, weer kunnen worden teruggelezen en op een willekeurig plaats op het scherm kunnen worden gekopieerd.

Muziek

Om de Plus/4 als een redelijke synthezister te gebruiken, heb je de beschikking over de nodige (Basic) hulpmiddelen. Het aansturen van de twee chips heeft niets meer te maken met een hele serie Pokes met moeilijk te onthouden waarden. Basic 3.5 staat wat dit betreft voor niets. Het volume is in te stellen met VOL, waarbij je 16 mogelijkheden hebt. Wil je een toon laten horen, dan is SOUND voldoende voor de Pus/4 om de noot te spelen. Voeg aan deze beide opdrachten wat data en een FOR/NEXT loop toe, en je Plus/4 speelt alles wat je maar wilt. De belangrijkste geluidsbewerkingen kent de Plus/4 dus, je zult daarentegen vergeefs zoeken naar opdrachten als ENVELOPE, VOICE en PLAY, waardoor het maken van muziek wel heel gemakkelijk zou worden. Maar dat is dan ook een van de weinige aanmerkingen die er op de Plus/4 Basic te maken zijn.

Meer struktuur

Bij de meeste Basic programma's is enige struktuur meestal ver te zoeken. De regels hangen aan elkaar van de GOTO's en IF's terwijl daardoor het zoeken van fouten juist zo moeilijk wordt. Als Basic programmeur kun je vaak niet anders, omdat er in Basic geen mogelijkheden aanwezig waren om een echte struktuur aan te brengen in de programma's. Commodore heeft dit gelukkig ook ingezien, en daarom een aantal opdrachten die een goede struktuur kunnen waarborgen aan Basic 3.5 toegevoegd. **IF..THEN..ELSE** en **DO/WHILE/ UNTIL** zijn daar de meest sprekende voorbeelden van.

Na een paar weken wat intensiever te hebben 'gespeeld' met de mogelijkheden die de C-16 (uitgebreid met RAM) en de Plus/4 te bieden hebben, moet ik bekennen, dat er veel meer mogelijkheden schuilen in deze zo verguisde huiscomputer. Al was het alleen maar vanwege het goed ogende toetsenbord van de Plus/4, of vanwege de veelheid aan effectieve Basic-opdrachten, de Plus/4 is werkelijk een volwassen computer, waaraan onterecht het label van ondeugdelijkheid en winkeldochter is toegekend. ●

L.Snel

De C-16 eigenaar, die op zoek is naar meer informatie, heeft het moeilijk. Toch zijn er meer dan tienduizend C-16 is ons land verkocht. Daarom dit artikel over machinetaal van Rob Vogelaar, die allerlei details heeft samengeraapt over de kleinste Commodore. Erg veel informatie en vrij diepgaand, maar wel de moeite waard, ook voor PLUS/4 bezitters.

Machinetaal C-16



De Basic 3.5 van de C-16 is voor allerlei tekentoepassingen erg gemakkelijk, maar na een tijdje wil je wat meer en dan ga je op zoek naar de Peek's en Poke's en de adressen en machinetaal. Het resultaat staat hieronder.

Adressen MAP C-16

$0000-$00ff =zero page
$0100-$01ff =stack
$0200-$0258 =basic input buffer
$0259-$0311 =ruimte om bijv. het middelpunt van een cirkel vast te leggen.
$0312-$0331 =ruimte voor b.p. vectoren o.a. IRQ ($0314,$0315)
$0333-$0334 =lo, en hi byte van het begin adres van het zojuist geladen of opgenomen programma.
$0335-$0336 =lo, en hi byte van het eindadres.
$0337-$034d =ruimte voor de file-naam.
$034e-$0472 =vrije ruimte voor korte routines.
$0473-$04e6 =verschillende charget-routines voor bijv. het lezen van de Basic tekst. Disassembleer maar vanaf $0473 dat is de charget die vergelijkbaar is met die van de C-64. Hierdoor is een wedge ook mogelijk.
$0500-$055e =ruimte voor de processor bijv. $0500 = $4c =(jmp), $0501, $0502 is de vector voor usr normaal staat daar $991c = "ill. quantity".
$055f-$0600 =ruimte voor de funktie-toetsen, de eerste 8 getallen zijn de lengtes van de acht verschillende opdrachten, de volgende plaatsen zijn de ASCII symbolen, probeer maar met het "m"- commando.
$0600-$0800 =vrije ruimte voor iets langere routines.
$0800-$0be7 =ruimte voor de kleuren van het scherm, hier kom ik nog op terug.
$0c00-$0fe7 =vrije ruimte voor de tekens op het scherm poke of "m" maar op 3072 of resp. $0c00.
$1000-$3ff6 =vrije ruimte voor het opslaan van de basic tekst en de verschillende variabelen.
$4000-$7fff =niet gebruikte ruimte voor het evt. uitbreiden van het geheugen doch de gebieden van $0000-$3fff, $4000-$7fff en $c000-$ffff stemmen met elkaar in, als je poked op $c000 of $0000, komt het ook op dat niet gebruikte gebied te staan.
$8000-$fcff =R.O.M. gebied met o.a. de interpreter en op $cdab staat nog een leuke routine, probeer hem maar uit.
$fd00-$ff20 =ruimte voor de T.E.D. chip en het i/o gebeuren op de T.E.D. chip, daar komen we nog uitgebreid op terug.
$ff81-$fff9 =de kernal sprong tabel identiek aan de C-64 en de VIC-20.
$fffa-$ffff =R.O.M. pointers voor NMI, RES, IRQ.
In het juli/augustusnummer 1985 van het duitse blad RUN staat overigens ook een uitgebreide Memory Map.

T.E.D. chip.

T.E.D. komt van Tekst en Display maar uit deze chip komt ook het geluid van de C-16. Hier volgen de mogelijkheden:

1. HIRES GRAPHICS.
2. MULTI COLOUR GRAPHICS.
3. NORMALE TEKST 40 kolommen, 25 rijen.
4. EXTENDED BACKGROUND COLOR.
5. MULTI COLOR TEKST.
6. USER DEFINED CHARACTERS.
7. REDUCED SCREEN 38 kolommen, 24 rijen.
8. A WHOLE SCREEN SHIFT IN 4 POSSIBLE DIRECTIONS (smooth scroll).
9. RASTER REGISTER (split screen), (more than one background color g.e. 10 background colors), (more than one character definition g.e. 768 possible different characters on a normal screen).
10. 8 VOLUME LEVELS.
11. 2 VOICES.
12. WHITE NOISE with voice 2.
13. SCREEN-BLANKING.

We gaan hieronder in op die mogelijkheden van TED.

1. Hires Graphics
Hires is 320 pixels horizontaal en 200 pixels vertikaal, het start adres voor de pixels is $2000-$3f3f. Verder geldt wat de pixels betreft hetzelfde als voor de C-64.
Het kleuren gebied loopt wat anders dan bij de C-64. Er zijn twee kleuren gebieden: $1800-$1be7 en $1c00-$1fe7, het gebied van $1800-$1be7 vertegenwoordigt de verschillende helderheidswaardes van de voor- en achtergrondkleuren, bijv. lokatie $1800 voor helderheidswaarde en knipperende kleureffecten.
Het gebied van $1c00-$1fe7 vertegenwoordigt de verschillende hoofdkleuren van de voor- en achtergrondkleuren, bijv. lokatie $1c00:
bit 7-4 = kleurwaarde voor de achtergrondkleur.
bit 3-0 = kleurwaarde voor de voorgrondkleur.
Hires is als volgt aan te zetten;
bit 5 van lokatie $ff06 moet een 1 bevatten, $ff06 bevat $3b normaal: $1b.
bit 2 van lokatie $ff12 moet een 0 bevatten, $ff12 bevat $c8 normaal: $cc.
Een machinetaalprogramma om hires in te schakelen kan er volgt uitzien:

```
a 1400 lda $ff06
a 1403 ora $20/uit-and $df
a 1405 sta $ff06
a 1408 lda ff12
a 140b and $fb/uit-ora $04
a 140d sta $ff12
a 1410 brk
```

Probeert u dit maar eens in de normale tekst mode, dus geen graphic statement gebruiken.
Als u dit heeft ingetikt en u "g"t het, dan ziet u dat u de kleuren met het tikken van letters veranderen. Het lijkt de C-64 wel en

dat klopt want het kleuren gebied staat nu van $0800-$0be7 6 $0c00-$0fe7 en dat is ook het tekst gebied. Als u hiermee kunt werken dan is er meer geheugen over want het gebied van $1800- $1fe7 wordt niet gebruikt.
Als u toch de kleuren in het normale gebied wilt hebben dan gaat dat als volgt:
in lokatie $ff14 staat het start adres voor de kleuren, normaal staat hier $0f maar omdat bit 3 altijd 1 is moet u dit lezen als $08.
Dit betekent dat nu alle tekst en kleur vanaf $0800 wordt gelezen, nu kunt u met het 'm' commando er $1f(=$18) van maken en u ziet dat u geen controle meer heeft over de tekst en kleuren. U kunt hieruit geraken door op gevoel in te tikken > ff14 08 en alles is weer normaal. Als u niet in machinetaal werkt en dus alleen maar met het graphic statement kunt u ook de tekst en kleurscherm verplaatsen door op geheugenplaats $07fb het juiste getal te plaatsen bijv. 08, normaliter staat daar natuurlijk $18.

❷ Multi-color graphics
Multi-color werkt net als bij de C-64.
Het is aan te schakelen door bit 4 van lokatie $ff07 aan te zetten d.m.v. ora£$10 en uit door and£$ef.
Hier volgen de verschillende kleuren registers:
Voorgrondkleur = $053b hier kunt u net als bij de C-64 poken (1339 decimaal) achtergrondkleur = $ff15 (65301)
Randkleur = $ff19 (65395)
Multi-color 1 = $ff16 (65302)
Multi-color 2 = $ff17 (65303)
Multi-color 3 = $ff18 (65304)

❸ Normale tekst
Hier volgen de verschillende registers met hun waarden:
$ff06 $1b
$ff12 $cc
$ff14 $08 ($0f)
$ff07 $08

❹ Extended background colormode
In deze weinig gebruikte mode heeft u vier verschillende background kleuren maar slechts 256/4= 64 characters.
U schakelt het aan met het zetten van bit 6 van lokatie $ff06, probeert u het maar eens, de cursor is weg maar typt u maar op goed geluk, maak eerst het scherm schoon.
Typ nu eerst een **a**.
Dan een **shift a**, dan in een **reverse (ctr19) a**, en daarna een **shift reverse a.** U ziet nu vier maal de **a** met vier verschillende achtergrondkleuren maar er zijn nog maar 64 characters mogelijk dus geen grafische characters meer, de 4 kleuren zijn resp. achtergrond en multi-color 1 t/m 3. U zet het weer uit door $1b in $ff06 te plaatsen ofwel:
aan = ora£$40
uit = and£$bf.

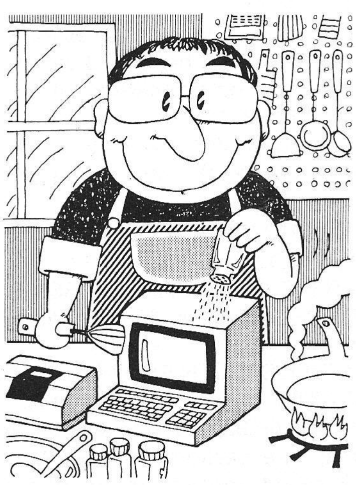
❺ Multi-color tekst
Dit kan ook en u hoeft alleen maar $ff07 weer in te vullen met $18 i.p.v. $08 zie ook punt 2.
$00 = achtergrond
$01 = multi-c. 1
$10 = multi-c. 2
$11 = voorgrond
Dit laatste geldt ook voor de multi-color graphics.

❻ User defined characters
Dit wordt bij de Commodores veel gebruikt, vooral bij de C-16 omdat deze geen sprites bezit.
Normaal haalt de C-16 de patronen voor de characters vanaf $d00- $d3ff, probeer het volgende maar:
Graphic 1,1
Monitor
t d000 d3ff 2000
x
graphic 1
Als u dit goed doet ziet u alle characters die de C-16 op het scherm kan zetten. Als we nu de characters zelf willen ontwerpen

Er is altijd interesse voor nieuwe of oude Peek en Pokes van Wim Wubs deze keer dit rijtje:

INTERESSANTE POKES EN PEEKES

PEEK (43)+256 ⋆ PEEK (44) begin basic programma
PEEK (45)+256 ⋆ PEEK (46) eind basic programma
PEEK (829)+256 ⋆ PEEK (830) begin programma met cass.rec.geladen
PEEK (831)+256 ⋆ PEEK (832) eind programma met cass.rec.geladen
PEEK (186) laatst gebruikt device
PEEK (144) moet na laden of saven O zijn(STATUS)
POKE 144,0 reset STATUS
PEEK (197) geeft waarde ingedrukte toets
POKE 788,52 run stop uitgeschakeld (normaal 49)
POKE 809,255 run stop restore + list commando werkt niet meer
POKE 774,40 na list regelnummer + list list enz.
POKE 774,0 na list alleen regelnummers
POKE 768,145 uitschakelen foutmeldingen
POKE 768,143 aan eind van programma automatisch koude start de scherm en tekst kleuren blijven echter gelijk (tevens direct- commando voor ontsnappen uit snellader)



POKE 212,1:inputa$ a$ is geopend met "(voor control commando's)
POKE 649,1 in combinatie met get-commando's zorgt ervoor dat bij indrukken van meerdere toetsen toch voor elke get-commando weer een toets ingedrukt moet worden.
PEEK (61731) is deze 85 dan is de betreffende computer een CBM- 64(voor het gebruik van een modum)
POKE 788,54 zet de klok stil (bij 49 loopt de klok verder)
POKE 56325,7 programma wordt vertraagd gerund(normaal 128)
Wanneer aan het begin van een basic programma POKE 788,52: POKE 768,143 gezet wordt is het programma helemaal beveiligd. ●

moeten we ze eerst in RAM zetten, een goede plaats daarvoor is $3c00. Eerst zetten we alle normale characters op die plaats met t d000 d3ff 3c00. Dit doen we omdat als we dadelijk gaan overschakelen naar dit gebied we eerst normale characters willen zien en er anders een wirwar in beeld komt. Voor het overschakelen is lokatie $ff13 verantwoordelijk.
bit 0 = altijd 1.
bit 7-2 bepaalt het startadres. Normaal staat hier $d0 (=$d1), als we hier nu d4 (=$d5) neerzetten schakelt het systeem over op $d400 en daar staan de kleine letters.
Stel we willen het richten op $3c00, dan moeten we eerst een bit op lokatie $ff12 clearen en dan kunnen we aan $ff13 klungelen.
Hier volgt het in assembly:

```
a 1500 lda $ff12
a 1503 and $fb      normaal or£$04
a 1505 sta $ff12
a 1508 lda $ff13
a 150B and£$03
a 150d ora£&$3c: het begin adres   normaal ora£&$d0.
a 150f sta $ff13
1 1522 brk
```

❼ Reduced screen

Dit betekent dat het scherm nu bestaat uit 38 kolommen en 24 rijen. Als u het scherm wilt reduceren naar 38 kolommen, dan moet bit 3 van lokatie $ff07 een 0 bevatten i.p.v. een 1. Moet het scherm 24 rijen breed worden, dan moet bit 3 van lokatie $ff06 een 0 bevatten i.p.v. een 1. Voor een normaal scherm vervangt u de 0 weer door een 1.

❽ Smooth scroll

Met Smooth Scroll kunt u het gehele scherm in 4 richtingen verschuiven en wel in 8 posities.
Naar links en rechts bit 0-2 van $ff07. Normaal zijn deze 3 bits 000, als u deze 3 bits ophoogt dus 001 dan verschuift het hele scherm naar rechts. Als u nog verder ophoogt schuift het nog verder. Wanneer de 3 bits 111 hebben bereikt is het hele scherm een characterplaats opgeschoven. Als u naar links wilt schuiven moet u dit register verlagen.
Naar boven en beneden bit 0-2 van $ff06. Normaal zijn deze 3 bits 011 en bit 3 = 1. Als u nu het scherm klein wilt hebben en de onderste rand bedekt, dan moet er in $ff06 $17 staan. Wilt u naar boven schuiven dan moet u het register verlagen tot $10.
Als u geen rand wilt bedekken en naar boven wilt schuiven dan moet er in $ff06 $1b staan en moet het register verlaagd worden.
Het schuiven in de y-richting levert wat problemen op maar U moet er maar mee experimenteren.
Als u een beetje in assembler kunt programmeren dan kunt u een routine schrijven die het hele scherm een bepaalde kant (links/rechts) op verplaatst. Als echter elke keer deze routine oproepen wordt, dan schuift het scherm zeer houterig. Door tussen deze routines 7 keer het schuifregister te verlagen of te verhogen, kunt u het hele scherm zeer smooth verschuiven.
Met behulp van deze techniek en raster technieken heb ik zelf een deel van een spel geschreven met op het bovenste deel van het scherm niet-schuivende letters en aan de onderkant d.m.v. user defined characters een door het beeld schuivend berglandschap. ●

De volgende keer gaan we verder met oa. Raster Interrupts en de verdere mogelijkheden van TED op de C-16 en Plus-4

Demodiskette-aanbod

De tijden van de maagdelijke diskettes in een doosje lijken voorbij te zijn. De leveranciers van schijfjes hebben ontdekt, dat hun medium zich heel goed leent om informatie te bevatten, voordat de eindgebruiker ermee aan de gang gaat. Noem dat reclame of demonstratie-software, de klant krijgt in ieder geval beschreven diskettes. 3M heeft nu een grote aktie gestart, waarbij van een doosje met 10 diskettes er al 5 vol staan met demo's. De meeste zijn voor de Commodore bezitter niet van belang, slechts wie een PC-10 of 20 bezit kan overweg met programma's als Framework, Multiplan en Travers.
Gelukkig is er toch ook nog aan de grootste groep computeraars in ons land gedacht en heeft Prompt Software uit Amsterdam een van de schijfjes in 64 format volgezet.
Deze firma heeft de ruimte op de schijf gebruikt om een soort catalogus van het software- en dienstenaanbod te geven.
Er staat ook een demonstratieversie op van het boekhoudpakket van Prompt, dat PC-Boek heet en in de echte uitvoering ƒ 999,- kost. Middels deze demo kan men toch een indruk krijgen van het pakket, dat de complete boekhouding voor kleinere en middelgrote bedrijven op een 64 mogelijk maakt voor een toch relatief bescheiden systeemprijs. Het pakket kan eventueel nog worden uitgebreid met een apart Verlies&Winst softwaredeel. Volgens Prompt is dit het best verkochte boekhoudpakket voor de 64.

PC beurs

Op 9 en 10 november is er in de Brabanthallen in Den Bosch een computerbeurs.